Rebuttal Report of James Bach on behalf of Hewlett Packard Enterprise Company

TurnKey Solutions Corp. v. Hewlett Packard Enterprise Co., No. 15-cv-01541-CMA-CBS (D. Colo.)

Signed: ______________________________
James Bach

Dated: February 3, 2017

Table of Contents

Introduction

This report is a rebuttal to the claims and arguments put forward by TurnKey regarding whether HPE misappropriated TurnKey's alleged trade secrets. I specifically considered TurnKey's various responses to interrogatories, the testimony of TurnKey witnesses, and TurnKey's Expert Disclosure by Dale Ellis (the "Ellis Disclosure").

I am being compensated for my time working on this matter at my standard rate of $300.00 per hour, plus reimbursement of reasonable direct expenses. I have no other interest in this litigation or the parties thereto.

I expect to be called to provide expert testimony regarding the opinions formed resulting from my analysis of the issues considered in this report. If asked to do so, I may also provide testimony describing the technology at issue, which I discuss below. I may rely on software user guides, manuals, handbooks, textbooks, technical literature, patents, presentations, software products currently and previously available to the public, my own personal experience with software testing products (including those relevant to this litigation) and in the field generally, and any other materials and/or information to explain the automated software testing technology at issue in this litigation. Additionally, I understand that I may be asked to provide a technical tutorial for the court and/or the jury. I also expect to provide expert testimony regarding the opinions and other matters set forth in this expert report.

I respectfully reserve the right to augment, modify, or supplement this report or my opinions, and/or to submit a supplemental expert report based on my continuing review of the evidence currently available to me, or in the event that I obtain any new evidence relevant to the opinions set forth herein, including additional deposition testimony, newly-produced documents, or new or rebuttal opinions offered by TurnKey's experts or witnesses for the first time after the submission of this report.

Additional Professional Experience Related To Automation[1]

From 1989 to 1991, I led the Special Projects and Methods team at Apple Computer, within the Development Systems Quality group. My team was responsible for developing test tools. In 1989, we created the TESLA tool, which was an early implementation of data-driven and keyword-based automation. We also created tools to support code coverage analysis.

When I moved to Borland International in 1991, I took over as test manager of the Borland C++ Debugger and Tools team. Our tools included performance testing and code coverage analysis tools. I also personally created tools to manage test assets for our department and to perform bug metrics analysis. At Borland, I had my first experience with an early product from Mercury Interactive called TestRunner. My team developed an experimental system, which ran on top of TestRunner, that we called an "object-oriented test automation" system. These days it would be recognized as keyword-based automation coupled with a domain-specific language. My colleague, Ramin Halviatti, took the technology further and patented it in 1995, after I had left the company.

Since founding Satisfice, Inc., I have frequently had clients who needed help with their automation strategies, or who hired me to build the automation for them. Those projects included:

[1] A more complete description of my relevant technical experience may be found in my Opening Expert Report dated January 3, 2017.

- **Satisfice.** Developed and distributed the *Allpairs* pairwise test generation tool, which creates a minimal set of test cases to cover all pairings of multiple variables.

- **Hewlett Packard.** Invented a methodology for documenting exploratory testing sessions called Session-Based Test Management (SBTM). Developed tools to manage ensuing test documentation.

- **Rational Corporation (later, IBM).** Provided consulting services to improve the RUP development methodology and tested the Rational Test Manager product.

- **Microsoft Antitrust Remedies Trial.** Performed a deep technical analysis of certain aspects of Windows XP on behalf of nine dissenting states, discovering evidence that convinced a judge to allow me to testify.[2]

- **Earthlink.** Developed an agile automation strategy based on small, inexpensive tools, with toolsmiths working closely with testers.

- **ManageMark.** Developed a Markov-model-based performance testing tool for online expense report management system.

- **FCC.** Developed tools to assist in testing bandwidth auction systems.

- **National Credit Union Association.** Demonstrated "headless" browser simulation to help test web-based services.

- **SearchFlow.** Developed a tool to test data integrity, and a tool that generated test cases from flowcharts.

- **BioSense Webster.** Developed systems to perform statistical analysis and visualization of test results for two different medical devices.

- **ADIA.** Developed extensive Excel macros to pull data from HPE ALM into spreadsheet-based reports.

- **QA Symphony.** Met with its development team and consulted on the design of features of its test recording tool that implement my SBTM method.

Background Of The Lawsuit

This is a lawsuit involving claims by TurnKey that HPE misappropriated its alleged trade secrets. Therefore, this lawsuit must be reviewed in the context of the software testing business. The software testing business, as it relates to the parties involved in this lawsuit, consists of the development of software tools that enable companies that use software in their businesses to test the software to insure as best they can that the software does what it is supposed to do.

[2] In this litigation, I filed a report but ultimately did not testify due to issues unrelated to the quality of my work. *See Microsoft Judge Indicates Tendency To Require Big Changes in Windows*, WALL STREET JOURNAL (May 8, 2002), *available at* http://www.wsj.com/articles/SB1020783398600636240; *Witness in Microsoft Case Is Dropped After Strenuous Protests by the Defense*, WALL STREET JOURNAL (May 10, 2002), *available at* http://www.wsj.com/articles/SB1020959298458433320.

Software testing consists of evaluating a software product[3] by learning about it through experimentation and exploration. Testing is distinct from *inspection*, which entails looking at the underlying "source code" and other constituent structures without actually operating the product. Software testers operate the software and evaluate products primarily for the purposes of finding what professionals in my industry refer to as "bugs"—which are errors, flaws, or failures that cause a software program to produce an incorrect or unexpected result, to behave in unintended ways, or to otherwise threaten the value of the product.

In order to perform the above activities, I and other software testers ultimately do two things:

1. **Exercise the product.** We click buttons, check boxes, access menus, enter data into text fields, click through different screens, and generally tour the interfaces of the product, supplying inputs or performing operations that elicit responses from the product. We also cause the product to be configured in different ways (via options or "preference settings," or by changing installation or environment states) that may affect its ability to perform. We do all this to simulate how a user might use the software product and ultimately push the product to perform all of the various operations that comprise it. In doing so, we input a large variety of data for the software to process.

2. **See what happens.** We observe and analyze the output, which, in the case of large software programs, can consist of an enormous and unwieldy amount of data. A large system is like a large city, bustling with activity, having many moving parts, and changing frequently. To test such a product can feel like touring a city—while trying to see everything that could go wrong with it.

Software testers need to perform tests quickly in order to find bugs, get them fixed, and get their software products out the door. This is why software companies develop or purchase tools that can ease the burden of simulating all necessary scenarios quickly while managing vast amounts of data. With these tools, we need not rely solely on humans to manually test the product.

The phrase "test automation" is somewhat of a misnomer, in that it is not actually about *automating* the test process. Testing, like programming, is something *people* do. Rather, "test automation" means people using tools to support the testing process. Test automation companies strive to understand the skills and needs and expectations of their customers. Responding to customer feedback is a major driver of test tool evolution.

Testers often wish to automate the "button pushing" part of the job. Thus, one popular approach to test automation is *user simulation*. In this form of automation, a tool operates a software product as a human user might. Tools directed at user simulation can be compared to self-driving tour buses that navigate through the various screens, features, and options of the software product undergoing testing (often referred to as an "application-under-test" or AUT). The tool periodically stops to check specific outputs; it can identify potential bugs when the output at a particular checkpoint is inconsistent with what was expected.

User simulation is the kind of test automation at issue in this case. Both HPE and TurnKey produce and sell such test automation tools to companies wishing to test their software.

[3] This includes software that is embedded in or interoperating with hardware products.

HPE's Test Automation Platform[4]

For purposes of this lawsuit, HPE develops and sells three products aimed at helping companies test their software.

- **Application Lifecycle Management ("ALM"),** which provides test management services. It consists of a database (or "data repository") that stores many different kinds of data—only some which are relevant to software testing—and a user interface.

- **Unified Functional Tester ("UFT," formerly known as Quality Test Professional or "QTP"),** which provides the capability of controlling (or "driving" as it is called) and scanning applications-under-test during the testing process.

- **Business Process Testing ("BPT"),** which is an add-on to ALM and UFT that provides keyword-based testing capabilities. It is uniquely configured for business software, such as software that facilitates the purchase of goods or services. Additionally, this particular product is the one that Turnkey alleges improperly utilizes its alleged trade secrets.

Taken together, these tools are a basis for developing, running, and managing keyword-driven user simulation of business software. They comprise a test automation framework, an ecosystem of interacting sub-systems. We also refer to HPE's test automation framework as a "platform" because it can accommodate products that use and extend its capabilities.

The following structures and terminology are relevant to HPE's test automation framework:

- **Object:** While this term has an important meaning within computing that is not applicable here, in the ALM world "test object" is a data structure that represents and contains information about some part of the user interface of an AUT. Types of objects include windows, text field, buttons, checkboxes, and menus, among other things. Objects are stored in HPE's ALM Object Repository.

- **Business Component:** A "business component" represents and contains information about a "business level" operation within an AUT.[5] Despite being called a "component," it is better understood as a process comprising a *set of steps*. These steps may be written as a low-level script ("scripted component"), or embodied within keywords that are associated with low-level scripts ("keyword-driven component").

  A business component typically refers to an entire screen or section of a screen, as well as to the theme and/or business processes associated with that screen, such as "create invoice," "place order," or "log in."

- **Flow:** A "flow" is a particular sequence of business components. In other words, it is a sequence of steps for interacting with the AUT, which is further composed of smaller sequences of steps that have been strung together.

- **Tests:** A "test" consists of information about one or more business components or flows or other scripts that relate to some evaluation of an AUT. This includes test data, which may be

[4] A more complete description of HPE's Test Automation Platform can be found in my Opening Expert Report dated January 3, 2017.

[5] The term "business component" was apparently invented by HPE, as this usage appears nowhere else in testing literature.

embedded in the test itself or stored in a data table of some kind (e.g. a data table from an external source, such as an Excel spreadsheet). A "business process test" is a test that consists of a sequence of business components and associated keywords, parameters, and data.

- **Keyword/Script Library:** A script or keyword "library" contains lists of scripts and/or keywords associated with scripts that may be used to describe business components, flows, and/or tests. When a keyword from the keyword library is invoked, the script corresponding to that keyword is retrieved from the ALM repository, along with the data from data sources linked to any associated parameters, and then executed.

- **Test Action:** These were introduced in v11.50 of UFT. Each action consists of its own test script (including all of the steps in that action), any objects in its local object repository, and any associated shared object repositories. An action is similar in function to a business component, but it belongs to UFT and does not require BPT.

- **Data Tables, Iterations, Configurations:** Data may be associated with components and tests in different ways. Data tables, iterations, and test configurations all involve the use of tables to store data that can be accessed by tests in different ways and at different times.

TurnKey's Software Testing Products

TurnKey, at all times pertinent to this litigation, developed and sold several add-on products designed to function with the HPE Software Testing Platform. By that I mean that Turnkey's products are intended to work with and call upon numerous functions provided by the HPE software platform in order to enable TurnKey's customers to utilize its automation tools, which TurnKey develops and sells under the names "cFactory" and "Accelerators."

- **cFactory:** cFactory is a tool for creating business components for HPE BPT. Through the use of programming interfaces provided by HPE, cFactory invokes the screen scanning capabilities of HPE UFT to collect information about an application that the user wishes to test. It analyzes this data using a special layer of code called a "rules base" which provides instructions for how to make business components in BPT that will simulate a user operating the application under test. After creating the components, the user can assemble them into business process tests using HP BPT.

- **Data Sheet Creator:** Having created the business process tests, the user invokes the Data Sheet Creator tool to create Excel spreadsheets that store test data. The Excel workbooks are stored in HP ALM.

- **dataDriver:** When the tests are actually run, dataDriver is the tool that controls that process. dataDriver reads the data in the spreadsheets and uses the data to control the flow of the tests.

- **Accelerators:** Accelerators are created for specific technology platforms, such as SAP or Oracle. Accelerators consist of test code (TurnKey calls it "content") in the form of components created by cFactory for objects commonly found in each platform. Accelerators are packaged work with dataDriver and Data Sheet Creator. The cFactory tool is technically required only to create new components or to update components.

I do not believe that there is a separate title that refers to all of the tools together. For the sake of simplicity, I will generally use the label "cFactory" to refer the cFactory tool, Data Sheet Creator, and dataDriver collectively. cFactory, in that collective sense, is the software product which TurnKey claims embody the trade secrets allegedly misappropriated by HPE.

HPE And Turnkey Entered Into A Business Relationship In 2012

I understand based on information furnished by HPE's counsel that HPE and TurnKey entered into a Software license and Distribution Agreement for Third Party Branded Products on May 16, 2012 (the "OEM Agreement"). I further understand that, from that time until sometime after TurnKey initiated this lawsuit, HPE sold TurnKey's add-on software tools, cFactory and Accelerators, alongside its own HPE Software Automation Platform.

The Gist Of The Lawsuit

Turnkey claims in this lawsuit that HPE tricked its employees into revealing trade secrets underlying its software automation tool, cFactory, which TurnKey had allowed HPE to sell pursuant to the OEM Agreement. TurnKey alleges that on two separate occasions, on October 7 and October 29, 2014, its employee Brad Kallaway revealed its alleged trade secrets to certain employees of HPE through a web-hosted teleconference (the "October Briefings").[6] TurnKey then alleges that HPE used those alleged secrets to develop a new version of BPT that HPE first announced to the market in December 2014 and officially released in September 2015.[7] TurnKey claims damages as a result of HPE's alleged trade secret misappropriation.[8] Based on the same general set of facts, TurnKey also brings claims for breach of contract and fraud.[9]

The Alleged Trade Secrets

As a matter of first principles, I began my investigation of TurnKey's allegations in this lawsuit by first trying to determine what Turnkey's alleged trade secrets are. I reviewed various formal statements made by TurnKey during the course of this litigation, including its responses under oath to HPE's interrogatories. I also reviewed the deposition testimony offered by numerous TurnKey witnesses, including TurnKey's Rule 30(b)(6) deposition specifically directed to the identification of Turnkey's alleged trade secrets. Finally, I reviewed the Ellis Disclosure, which details the alleged trade secrets as claimed by TurnKey. This amounts to a lot of text, much of it redundant and convoluted. Therefore, I undertook to restate TurnKey's alleged trade secrets in a more compact form in this report.

However, my job was made more difficult by TurnKey's failure to provide any reasonably specific description for its alleged trade secrets beyond what I have described here. Although Turnkey has provided substantial text and testimony, the information therein (1) is merely suggestive without being definitive, (2) merely states explicitly what is common knowledge to anyone having ordinary skill in the art, or (3) merely re-states information about its products which are widely and routinely disseminated to the public by TurnKey.

TurnKey has complicated these efforts even further with its ever-changing list of what the alleged trade secrets actually are. Although TurnKey's Complaint and interrogatory responses identify five

[6] Complaint ¶¶ 39-48.
[7] Complaint ¶¶ 49, 59-64.
[8] Complaint ¶¶ 65-68.
[9] Complaint ¶¶ 69-79.

alleged trade secrets,[10] TurnKey's expert reports and disclosures identify only three, combining some alleged secrets and omitting others.[11] For the purposes of this report, I rely primarily on TurnKey's Supplemental Response to Interrogatory No. 1, because that seems to be the most direct and succinct statement of TurnKey's position, supplemented by the details and explanations found in the 30(b)(6) deposition of Dale Ellis.

In my opinion, every description below represents the essence of TurnKey's alleged trade secrets based on TurnKey's own testimony and written statements. In other words, as an expert in the field of testing and automation used for testing, and having reviewed all the materials in this case, I believe that the short statements below represent everything definitive about the alleged trade secrets actually asserted by TurnKey.

TurnKey has alleged five trade secrets:

1. Auto-Parameterization
2. Auto-Generation of a Data Repository
3. Automatic Detection of Changes in a Business Application and the Updating of BPT Components
4. Automated Component Generation
5. Proprietary Keyword Methodology/Framework

Auto-Parameterization

TurnKey claims that auto-parameterization is a process that occurs during the automated creation or maintenance of a business component within HPE ALM using the open API provided by HPE, each time a step (which consists of an object, operation,[12] and value) is added to that business component. [redacted]

- [redacted]
- [redacted][13]

The following notes are important for interpreting TurnKey's descriptions of this alleged secret:

[10] Complaint ¶ 46; Turnkey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016). TurnKey's Rule 30(b)(6) deposition on the topic of TurnKey's trade secret designations is consistent with the list of trade secrets articulated in these written documents. *See* Ellis 30(b)(6) Deposition (Feb. 24, 2016).

[11] *See* Ellis Disclosure, at14 (Auto-Generation of Test Components/Autoparameterization) and 19 (Data Repositories); Pedigo Opening Damages Report (Jan. 3, 2017), at 4-5.

[12] TurnKey uses the term "action" for this, but the more correct term in BPT is "operation." Test actions are a slightly different thing on the HPE platform.

[13] *See, e.g.* TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016), at 2.

- In computing, a parameter is just a temporary container for input or output. TurnKey's Supplemental Response to Interrogatory No. 1 does not specify that the parameter *must* refer to a table (it states that "a parameter can comprise a name of a column in a test data sheet"). Therefore, I infer that its description of auto-parameterization includes all kinds of parameters, including those that refer to simple (non-tabular) variables.

- TurnKey's own product requires a separate, manually initiated step to link the parameters in its objects to a table: the user must invoke the Data Sheet Creator tool. Thus, the auto-parameterization process is not fully automatic. Therefore, I infer that auto-parameterization as described by TurnKey may involve some manual intervention.

- As I will discuss below, TurnKey has been inconsistent about whether capturing data about screen objects is included in auto-parameterization. Based on my knowledge of the art, I conclude it is not included.

Auto-Generation Of A Data Repository

TurnKey claims that auto-generation of a data repository is a process initiated by the user after the user has created a test case in HPE ALM that includes parameterized business components. [redacted][14]

Automatic Detection Of Changes In A Business Application And The Updating Of BPT Components

TurnKey claims that automatic detection of changes in a business application and the updating of BPT components is the process of identifying changes in a business application-under-test that affect BPT components, and providing a facility for user review and acceptance of those changes. [redacted].[15]

Automated Component Generation

TurnKey claims that automated component generation is the automated process of creating BPT components with appropriate steps and parameters within HPE ALM. [redacted]

- [redacted]

- [redacted]

- [redacted]

[14] *See, e.g.*, TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016), at 4.
[15] *See, e.g.*, TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016), at 5.

[redacted] [redacted].[16]

Proprietary Keyword Methodology/Framework

TurnKey does not use the term "methodology" consistently in its statements. But it seems to claim that its proprietary keyword methodology/framework could consist of *any of* four things:

1. The *software code* embodying specific algorithms that constitute the TurnKey product, including keyword libraries, rules bases, and all other code in the product.[17]

2. The *plainly visible behavior* of Auto-Generation of Components, Auto-Parameterization, Auto-Generation of Data Repositories, and Auto-Detection of Changes as they operate together in TurnKey's product.

3. The *ways users interact* with a specific product that performs Auto-Generation of Components, Auto-Parameterization, Auto-Generation of Data Repositories, and Auto-Detection of Changes.

4. The *design methods* by which TurnKey personnel develop its keyword libraries and rules bases.

None Of The Information Alleged By TurnKey To Describe Its Alleged Trade Secrets Qualifies As A Trade Secret

It is my understanding that for information to qualify as a trade secret under Colorado law, it must meet these conditions:

1. It must be specified with reasonable particularity.[18]

2. It must not already be generally known within the industry.

3. It must derive value from its secrecy.

[16] *See, e.g.*, TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016), at 6.

[17] *See, e.g.*, TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016), at 6-7.

[18] *See, e.g.*, *Saturn Sys., Inc. v. Militare*, 252 P.3d 516, 522 (Colo. App. 2011) (citing to *Imax Corp. v. Cinema Techs, Inc.*, 152 F.3d 1161, 1164-65 (9th Cir. 1998) for the proposition that the plaintiff in a trade secret case should describe the subject matter of the trade secret with sufficient particularity); *L-3 Commc'ns Corp. v. Jaxon Eng'g & Maint., Inc.*, No. 10-CV-02868-MSK-KMT, 2011 WL 10858409, at *1 (D. Colo. Oct. 12, 2011) (a plaintiff "will normally be required first to identify with reasonable particularity the matter which it claims constitutes a trade secret, before it will be allowed ... to compel discovery of its adversary's trade secrets") (quotation omitted); *id.* at 3 ("[G]eneral allegations and generic references to products or information are insufficient to satisfy the reasonable particularity standard.") (quoting *Hill v. Best Med. Int'l*, No. 09–1184, 2010 WL 2546023, at *4 (W.D. Pa. June 24, 2010); *IDX Sys. Corp. v. EPIC Sys. Corp.*, 285 F.3d 581, 583-84 (7th Cir. 2002) (trade secret description inadequate if it is overly broad or general); *id.* (trade secrete description inadequate if it refers to documents without specifying what information in the documents allegedly constitutes the trade secrets); *Sit-Up Ltd. v. IAC/Interactive Corp.*, 2008 WL 463884, at *11 (S.D.N.Y. Feb. 20, 2008) (trade secret description inadequate if it describes an overall business method or process (without specifying which parts are secret).

SmartBear and Ranorex. SAP also has its own tool called CBTA, which performs the learning, auto-generation and auto-parameterization steps, at the very least.[50]

In my opinion, TurnKey's claimed "trade secrets" are simply variations of longstanding industry solutions. I understand that another expert, such as Mr. Ellis, might disagree with me on my analysis, but not without taking reasonable diligence to review the history of the art.

Turnkey's "Trade Secrets" Are Largely Self-Revealing Or Have Been Publicly Disclosed In Turnkey's Marketing Efforts

As a programmer and tester, I am used to dealing with deceptively complex things. The code that runs our world behind the scenes is enormously, breathtakingly complex. I once consulted at Volvo, where I learned that it takes millions of lines of software code to run a car these days, performing all manner of obscure functions. Cars look simple on the outside, but they are complex on the inside—that is okay, because drivers and passengers do not need to know how it all works.

By contrast, the functions and capabilities of cFactory are in plain view. They harbor little hidden complexity, and if that were not true *TurnKey could not sell it*. Although TurnKey's source code is hidden from users, nothing else is. cFactory's primary capabilities are and *must be* visible to users in order for that software to have any value in testing. This is true for two reasons:

- We testers need our test tools to be open and predictable because when we test a product, we must know exactly how it is being exercised. We can't let operation and verification steps be magic. We cannot test by faith. Testers who use automation tools are not merely button-pushers.

- cFactory is a tool that works exclusively within the HPE automation platform. It is an add-on to HPE ALM, BPT, and UFT which makes heavy use of the HPE automation platform's capabilities. cFactory provides several features that extend HPE technology. All of those features and all of the output that results from those features are necessarily stored in the ALM repository, unencrypted and in full view.

Therefore, when I say that TurnKey's alleged secrets are self-revealing, I mean that cFactory, by its nature, possesses certain capabilities that must be shown to, explained to, understood by, and employed by anyone who seeks to use the product for its intended purpose, and that the "secrets" which TurnKey has asserted against HPE are nothing more than those very capabilities. *To describe the important things that cFactory does is simply to describe what any ordinary, skilled user of BPT does.*

Given the nature of the "secrets" TurnKey has asserted, it is impossible to protect them if anyone outside of TurnKey is allowed to see the product work. If TurnKey really wanted to keep them secret, it would have had to operate as a consultancy rather than a software product vendor. As a consultancy, it could build cFactory in complete shadow, operate it exclusively with its own employees, apply it privately to client projects, and then it could sensibly claim that every basic capability of cFactory is a trade secret. But that is not what TurnKey has done here. Instead, TurnKey conducts public demonstrations of cFactory (including demonstration videos to YouTube), and markets and sells cFactory to any willing customer. My analysis of Mr. Kallaway's statement of

[50] *Component Based Test Automation How-To Guide* (Feb. 2012), *available at* https://goo.gl/qhjY1M.

what is proprietary about cFactory demonstrates how his statements mirror TurnKey's public statements and demonstrations.[51]

This is what TurnKey publicly says about how open and non-mysterious its product is:

> What we do is we provide a prebuilt list of rules for automating web Java, .net, SAP, and so on, different applications, but these rules are completely configurable so that you can say, 'OK, we're using this particular widget from this vendor,' and we've built it into our application. 'Whenever you, cFactory, see this widget on the screen, here is what I want to do. These are how I would like to manipulate this object. I want to tap through the values or I want to input data or I want to validate the data' **so that these components that cFactory creates are building components the way you would build them if you were doing it manually. As you're going to see, there's really no magic to this.** It's all a very straightforward solution in that you're going to tell it, 'Here's the application I'm testing and here's how I want to test it,' and cFactory, on clicking the screens, creates these components that become part of your component library so that creating a test-- remember what we did is we created a component for each screen-- is simply identifying the screen flow….
>
> **In fact, cFactory, once you create the components, that's it.** It's not adding any overhead. cFactory creates the components. **The components go into Quality Center, and from then on it's Quality Center and QTP.** CFactory, for all intents and purposes, is a desktop app. It is licensed as a concurrent user application. Most customers only need one, maybe two copies of it. Because once you create these components and drop them into the business process repository, then anyone who has got a BPT license can go in, drag and drop these components, create tests cases, and execute them, and create the Excel spreadsheets.[52]

In other words, cFactory does not optimize or improve upon what ordinary users of HPE technology already do. It does the *same things* that they do. For the same reason that the inventor of a bread-baking machine cannot claim that "baking bread" is a secret, TurnKey cannot claim that scanning a screen, collecting objects, and turning them into components is, as such, a secret. These things have value, being automated, but that value to the user has absolutely nothing to do with them being special, different, and not publicly known. If, however, TurnKey's alleged secrets have something to do with innovative and secret algorithms, rather than behavior that is strictly in plain view, then it must disclose what those algorithms are. It has not done so.

Related to this is the issue of "similarity."[53] Given that cFactory is a product developed to perform tasks that HPE users already perform, it is not surprising that there would be similarities between the TurnKey's technology and HPE technology. TurnKey has intentionally fostered that similarity. Almost 40% (127 out of 331) of the figures in the cFactory User Guide are screenshots of *HPE products*, and some of the user interface of cFactory appears to be designed intentionally to mimic the look and feel of ALM/BPT/UFT. TurnKey speaks of this closeness publicly as a virtue:

> "Dan Gannon: We get this question from customers in the buying process often where they say, 'Well, what happens if we want to get rid of you?' We say, 'Well, you can. The assets you have built and developed are all HP standard, and you can continue to use them for forever.' **There's nothing TurnKey about the BPT component. That's a standard HP construct.**"
>
> "Bernard: Are the tests and datasheets -- the Excel workbooks -- stored in HP ALM as well? Dan Gannon: They absolutely are. **We want to leverage ALM to the greatest extent possible. QC is a remarkable application type. We use that as our framework for**

[51] *See* Appendix A, at A36.

[52] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

[53] *See*, *e.g.,* Ellis Disclosure, at 9.

> **everything we do. You're going to store these projects inside of ALM. The datasheets are stored inside of ALM.** You have all the security provisions that govern that as well. All of that, once you're finished, building components or maintaining components, you're operating everything inside of ALM. **It really helps from a training perspective because you're going to do the same things you do today**.
>
> "Dan Gannon: When we built our product, we built it from day one completely wrapped inside of ALM QC. It isn't just a light integration, but we're completely intertwined and baked in. **The term integration probably doesn't really do justice to how tightly connected we are with ALM.** As I mentioned at the outset, our products don't even work without that platform in place. There's a real, real tight, tight tie."[54]

This closeness means that it should not surprise anyone that product improvements in HPE's platform may occur in areas that cFactory also serves, through the natural process of its independent development efforts.

Algorithms And Code Might Be Trade Secrets, But They Are Nowhere To Be Found In TurnKey's Disclosures

Code is the set of exact instructions that a computer follows. An *algorithm*, broadly speaking, is a "series of computational steps."[55] An algorithm need not be specified in the form of executable code, but it must be specific enough to allow code to be written without requiring creative interpretation.

TurnKey has identified neither algorithms nor code in connection with its alleged secrets. This matters because TurnKey is neither specific nor explicit about how its alleged secrets can be implemented, making that information unusable to HPE. Instead, TurnKey discloses only that each one depends on some unspecified "intelligence"[56] to get the job done. Therefore, whatever TurnKey conveyed to HPE about its alleged trade secrets could not be programmed into a computer. The valuable part of each alleged secret was omitted.

As I will demonstrate below, not even hundreds of questions during depositions could coax these algorithmic details from TurnKey personnel.

TurnKey does not dispute HPE's claim that the actual code implementing ALM, BPT, and UFT was written by HPE personnel. Yet nowhere in the Ellis Disclosure does Mr. Ellis explain any theory for *how* TurnKey's alleged secrets actually made their way from the mind of Brad Kallaway to the minds of HPE's employees in less than an hour,[57] and then in the months that followed, somehow became transformed into explicit, rigorous software code.

In contrast, HPE *does* share source code and specific algorithms to allow TurnKey and other parters and vendors to perform the same kinds of things that TurnKey claims as trade secrets. For example, HPE provides an Open Testing Architecture API which enables vendors like TurnKey to access the services and capabilities of HPE technology. In fact, most of the alleged secrets that TurnKey describes rely integrally on HPE to do the heavy lifting.[58]

[54] Daniel Gannon, *Transforming Test Automation Scriptless Testing Comes of Age*, YOUTUBE (March 5, 2015) *available at* https://www.youtube.com/watch?v=jeb9-MrzLMs.
[55] *See* Weinberg, et al., *Computer Information Systems: An Introduction to Data Processing* (1985).
[56] See my analysis of Brad Kallaway's use of the term "intelligence" in Appendix A, at A36.
[57] See my derivation of this time period in Appendix A, at A4.
[58] *See* Appendix C for details.

TurnKey met its burden of proof, and that, in fact, copious documentary evidence proves that TurnKey allegations are without foundation.

Analysis Of Alleged Trade Secret #1: Auto-Parameterization

This Alleged Secret Was Not Specified With Reasonable Particularity

TurnKey's descriptions of this alleged trade secret have been inconsistent and vague.

Auto-Parameterization Is Described Differently By TurnKey At Different Times And Places

Based on TurnKey's attempted explanations of this trade secret, the scope of "auto-parameterization" as claimed by TurnKey is unclear. What does it include and what does it not include? For example, does the auto-parameterization process include capturing data from the application screen? Does it specifically include the particular form of capture called "scanning?"

It seems to me, as one skilled in the art, that capturing information about the application screen is *not* part of auto-parameterization. And indeed capturing data is not mentioned in the original complaint:

> Autoparameterization: among other capabilities, this process auto generates parameters while creating automation steps as a part of a BPT components and enables users to define data outside of the business component. It is one of the most innovative and valuable tools incorporated into TurnKey's cFactory™ product.[94]

TurnKey changed this description substantially in its Supplemental Response to Interrogatory No. 1.

> [redacted].[95]

By this point, I was no longer sure whether TurnKey considered capturing data from the screen to be a separate activity because "analyzing controls on a screen" might include the process of capture, or it might only refer to the analysis occurring *after* information about the controls has been captured.

Notice further that neither of the above descriptions says anything about "scanning" versus "recording." TurnKey's interest in that distinction is a more recent development, indicating that it is still uncertain of what it is about auto-parameterization that it can claim to have invented.

For instance, in his Rule 30(b)(6) deposition testimony on behalf of TurnKey, Mr. Ellis gave a clear answer about TurnKey's position on this question: not only is screen object capture not a part of auto-parameterization, but specifically *scanning* is not:

[94] Complaint ¶ 46.
[95] TurnKey Supplemental Response to Interrogatory No. 1.

> [redacted][96]

That seems to match the description of auto-parameterization given by TurnKey's damages expert in his opening damages report, which similarly does not include the capture process:

> [redacted][97]

But Mr. Ellis has reversed himself in his own report, which explicitly includes scanning the screen as a part of this alleged secret:

> More specifically with respect to the technical aspects of the overlap, **cFactory's methodology for accomplishing autoparameterization involves invoking the object scanning capabilities of UFT to scan all or a region of an application screen of an application under test ("AUT");** capturing relevant information about the objects in an object repository; using that information to identify the objects for which a test step (part of a BPT test component) should be created; using logic pre-programmed in the software to analyze the data scanned and captured to determine the type and number of parameters to be used in a particular step based on the testing action that has been identified from the learned characteristics of the objects; determining the appropriate names for selected parameters by analyzing the structure of the application screen as reflected in the learned characteristics of the objects; and automatically inserting the appropriate parameter information in the test step that helps comprise the BPT test component.[98]

Mr. Ellis further claims that what makes BPT Version 12.5 misappropriate TurnKey's alleged auto-parameterization trade secret is that it now supposedly does scanning and not just recording:

> The methodology by which this BPT v. 12.5x autoparameterization is accomplished is substantially similar to the TurnKey methodology revealed by Mr. Kallaway to HP in October 2014. **For one thing, it is not done through a recording methodology. Rather, HP uses the scanning capabilities of UFT to scan and capture relevant data regarding the objects on an application screen or region of a screen and store that information in an object repository.**[99]

So now, TurnKey's position is that, not only capturing, but specifically "invoking the object scanning capabilities of UFT" is part of their secret. They claim that they gave the idea of scanning to HPE in confidence. Consider the absurdity of this claim: HPE created UFT with the capability of scanning an application screen. TurnKey duly accesses this capability, as specifically contemplated and enabled by HPE itself, via the Automation Object Model API. At what point in that process is *anything* about the process a secret held by TurnKey? How can TurnKey argue that HPE is not permitted to use its own invention for the exact purpose for which it was designed?

[96] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 28.
[97] Pedigo Opening Damages Report (Jan. 3, 2016), at 5.
[98] Ellis Disclosure, at 14.
[99] Ellis Disclosure, at 17.

In my expert opinion, Mr. Ellis was correct in his original testimony: object capture (including scanning) is not part of auto-parameterization. Whether steps are auto-parameterized as a batch, or one-by-one, makes no difference to the process of auto-parameterization.

TurnKey's Description Of Auto-Parameterization Is Vague And Non-Technical

The first sentence describing auto-parameterization in TurnKey's Supplemental Response to Interrogatory No. 1 communicates nothing actionable or computational:

> [redacted][100]

In this description, we find these verbs: *creates, analyzing, learn, associating, creating, generating,* and *assigning.* What we don't find is any description or explanation or algorithm to tell us what's behind those verbs. This description is not telling me how any of this is done; it is simply saying that something, somehow, someway, is done. A non-technical person might read this and assume that a person of skill in the art would "just know" what this means. I am a person with skill in the art, and I am certain it contains no significant content. This text might be *referring* to a technical secret. It is most certainly not describing one, and no one who is given this description would be enabled to build anything they he did not already know how to build.

The rest of the description continues in a similar vein. I reviewed it, and I do not see technically actionable content.

Auto-Parameterization Is Not Described in The Slides For The October Briefings

If the essence of the secret to auto-parameterization is not found in the Supplemental Response to Interrogatory No. 1, could it be elsewhere?

The slides that TurnKey used during the October Briefings were marked confidential, suggesting that they might contain something secret.[101] Is the auto-parameterization trade secret in there?

> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]

[100] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).

[101] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at Ex. 4.

> [redacted][102]

By TurnKey's own admission, the slides from the October Briefings did not describe TurnKey's alleged auto-parameterization trade secret.

Mr. Ellis' Description Of Auto-Parameterization Is Vague And Non-Technical

If TurnKey's alleged auto-parameterization trade secret is not described in the slides from the October Briefings, can Mr. Ellis tell us what it is?

> [redacted][103]

Recall that TurnKey claims that this secret is so obscure and difficult that HPE personnel could not reasonably have been able to develop it independently. So, I am surprised that Mr. Ellis believes he can describe the technical details of an important technical secret in only ten minutes, without reference to diagrams, documentation, or software code.

This is exactly what Mr. Ellis says about TurnKey's auto-parameterization "secret" in his Rule 30(b)(6) deposition on that very topic:

[102] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 43.

[103] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 46.

Location		Testimony[104]	My Paraphrase
[redacted]	[redacted]	[redacted]	[redacted]

[104] Ellis Deposition (Feb. 24, 2016). The page numbers and line numbers of the testimony cited in this table are provided in the first and second columns, respectively.

This describes no algorithm. This describes no specific operation. It does not mention "scanning." The process described here sounds like little more than using an index to access information stored in a file. As such, this is ordinary computing practice, and not anyone's trade secret. Moreover, based on other testimony, we know that *what Mr. Ellis described is not actually what cFactory does*. [redacted][106] but Mr. Ellis doesn't tell us that. He also doesn't tell us how the [redacted] code is executed. The burden is on TurnKey to say precisely what its auto-parameterization trade secret is, but major elements of that process are left undescribed. Dale Ellis says:

> [redacted]
>
> [redacted][107]

But this is not an "example" of a rule. It is a description of *what a rule can do*. We are left to guess how that rule is implemented.

The auto-parameterization idea described here could be implemented in many different ways. Anyone who wanted to recreate this technology would have many choices to make and a great many details to fill in. That is a major engineering undertaking.

In my experience, a reasonably competent software developer would need to ask the following questions—all left unanswered by Mr. Ellis—in order to determine how to implement auto-parameterization in the manner used by TurnKey:

- [redacted][108] Is that used to execute the rules base code?
- How specifically do you integrate [redacted]into your product to make the rules base work?
- In what format do I receive the data about the objects in the application?
- How do I feed that data to the rules base?
- What output comes from the rules base? What format is it in? XML? JSON?
- How do I act on that output to retrieve the actual parameters and names?
- If the renaming is mediated by the rules base, and only the rules base, why wouldn't the rules base choose correct names on the first pass? Under what specific circumstances does anything need to be renamed, and when would the rules base be invoked again to do that?
- If the renaming is a separate process from that which the rules base performs, what is that process?

[105] Kallaway Deposition (May 17, 2016), at 17.
[106] Jacques Deposition (June 8, 2016), at 83.
[107] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 108.
[108] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 46.

- Does auto-parameterization also deal with output fields? How does that work?
- Does the order of the objects in the data I receive from UFT matter? If so, in what specific way?
- Does the proximity of the objects in the data I receive from UFT matter? If so, in what specific way?
- Does the rules base [redacted] code get executed once for creating components and then a second time for getting the parameters or is it all one process?
- Does the rules base create many components at once when its code is executed or do we call it one at a time for each component?
- If a component depends on an object that is off screen or invisible, can the rules base handle that? If so, in what specific way?
- Can the rules base handle situations where the same object is shared by multiple components? If so, in what specific way?
- What kind of test data or scenarios do you suggest for testing auto-parameterization?
- In what specific way can rules relate objects? What semantics are available within the confines of a rules base to handle unusual or especially complicated clusters of objects that make up a component?
- Are there any special libraries or APIs that I need to use to make your version of auto-parameterization work? What are they?
- What error conditions do I need to handle?
- How do I write the code to manage performance and resource issues?
- Are there any kinds of objects that your technology is not able to handle? If so, why?
- What specific design guidelines do you use for architecting a good set of rules?
- There are lots of ways that login windows and dialog boxes might be designed and situated. Does your solution reliably handle all of them? How specifically did you design your rules to accomplish that? Or did you just design something that works for some things, some of the time?

It would not have been possible to give technically operational answers to these questions in the time Mr. Kallaway had to do his briefings—and these are just my questions for one of the secrets.

Yet these questions relate to the problem of doing actual software development. Software developers do not build software products based on five sentences of high-level product description. This is why it is my opinion that if there were a genuine secret here, it would be in the source code. As explained above, Turnkey's claim cannot be based on its source code, since TurnKey's source code was never turned over to HPE, never shown to HPE, and never made available for HPE to analyze prior to this litigation.

Mr. Ellis provided five sentences of content which disclosed no specific algorithm, formula, or revelation that could have been appropriated. He described no specific rules, nor even the format or semantics of rules in general. He says something about renaming parameters, but explains nothing about how it works. If Mr. Ellis' description represents the level of specificity that was given to HPE in the briefings, then the briefings were in no technical sense a "deep dive."

Mr. Kallaway's Description of Auto-Parameterization Is Also Vague and Non-Technical

During his deposition, Mr. Kallaway provided fifteen minutes of testimony in an attempt to explain what is secret about cFactory.[109] Based on my analysis of Mr. Kallaway's deposition transcript, it appears that his statements are no more technical or specific than TurnKey's public statements about cFactory.[110] This is important because TurnKey claims it is Mr. Kallaway who conveyed the alleged trade secrets in the October Briefings.[111]

For instance, Mr. Kallaway testified: [redacted][112] This sounds just like what Becky Karch said in a *public* TurnKey webinar: "[cFactory] parameterizes and assigns keywords to each of the fields that you see..."[113] I am not cherry-picking. Nowhere in Mr. Kallaway's testimony does he provide any significant detail about the processes of auto-parameterization.

Despite numerous opportunities to describe its alleged auto-parameterization trade secret in detail, TurnKey has failed to do so with reasonable particularity.

This Alleged Secret Was Already Known To The Industry

Auto-parameterization is not a secret.

TurnKey did not invent auto-parameterization. Auto-parameterization is just parameterization that has been automated in some way.[114] That is all. Instead of requiring the user to create parameters by hand, a tool performs this action. Generically, auto-parameterization is a two-step process:

1. Identifying variables associated with a data entity of interest.

2. Assigning to that entity parameters which correspond to the identified variables, such that the values of those variables may be accessed or assigned via those parameters when working with the entity.[115] (This access or assignment process may take different forms in different contexts. One example of assignment could be parameters linked to an internal or external data source. Another example could be the ability to edit code containing parameters for a "function call" that invokes the desired entity.)

[109] Kallaway Deposition (May 17, 2016), at 16.
[110] My analysis of Mr. Kallaway's deposition transcript is set forth in further detail in Appendix A, at A26-A27.
[111] Ellis Disclosure, at 3.
[112] Kallaway Deposition (May 17, 2016), at 18.
[113] *Developing a Comprehensive Test Automation QA Strategy*, YOUTUBE (Aug. 8, 2015), *available at* https://www.youtube.com/watch?v=TVshi26PItk.
[114] For further information about parameterization, see my Opening Expert Report, at 9-10.
[115] I base this description on the general computer science meaning of the term "parameter," as exemplified in *International Standard: Programming languages—C*, § 3.16, ISO/IEC 9899:201x (Committee Draft) (Apr. 12, 2011), *available at* http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1570.pdf.

> checks and actions in the action list with a user interface element of an application to be tested. For example, **test data defines data that is to be input into a user interface element of the application**. The value to be input is stored along with a set of characteristics or properties of the user interface element the data is to be input into, such as a type of the user interface element (e.g., a text input field), a size, identification information (e.g., field name or table name, position in hierarchy of screen objects, etc.) and similar user interface characteristic or property information."[121]

In this invention, test data is separated from the objects and the steps to be performed. The test data is stored in a separate table. Then the test data is reunited with the step to be performed at the time the test is executed. This is parameterization, and it is based on logic encoded separately from the generic master test script.

In addition to these examples, at last year's StarEast conference,[122] I also witnessed demonstrations of products from Ranorex and Tricentis Tosca that have similar capabilities.

There are lots of ways to do auto-parameterization and to package it. It is in no way an unusual or secret concept. I see no meaningful difference between TurnKey's vague and general specification of its secret and any of the examples I've just described.

This Alleged Secret Derives No Value From Secrecy

If TurnKey were to hide every aspect of auto-parameterization from public view, it would not impair any other vendor from implementing this technology. I know this because the technology needed to implement every aspect of auto-parameterization is already provided by HPE itself, as part of its Open Test Architecture API, which is exactly the means by which cFactory does auto-parameterization.

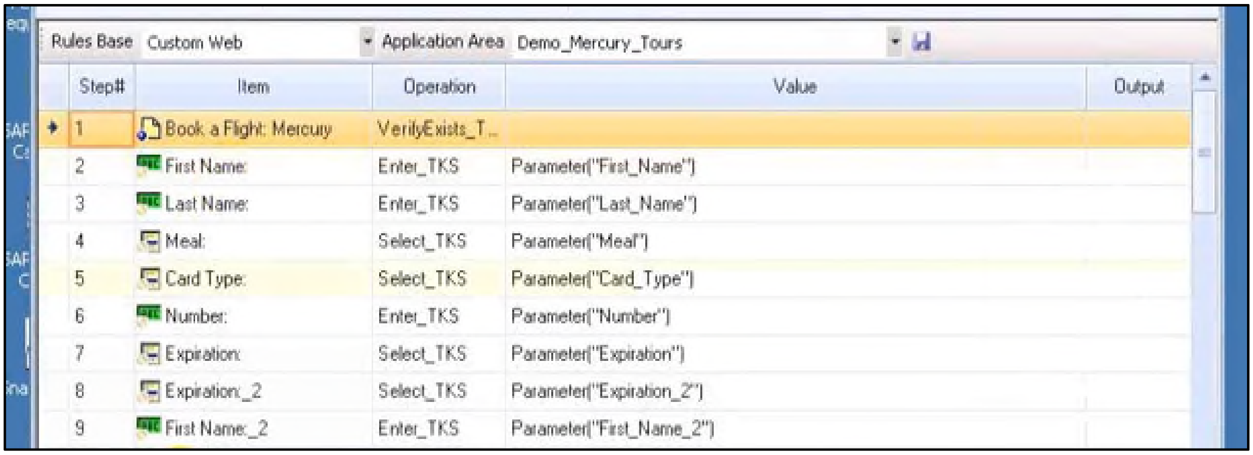
The only aspect of TurnKey's auto-parameterization "methodology," which in my opinion might have value that derives from secrecy is the contents of the TurnKey rules bases, as a whole. Not the *concept* of rules bases (or any other high-level overview), but rather the code itself. However, TurnKey has not identified its rules base as a trade secret, nor has it alleged misappropriation of its rules base.

This Alleged Secret Was Not Kept Secret By TurnKey

TurnKey does not hide the auto-parameterization functionality. Here are screenshots from a public demonstration of cFactory:

[121] U.S. Patent No. 8,66,467 (column 2, line 55).

[122] StarEast and StarWest are two of the largest industrial testing conferences in the United States.



Rules Base: Custom Web
Application Area: Demo_Mercury_Tours

Step#	Item	Operation	Value	Output
1	Book a Flight: Mercury	VerifyExists_T...		
2	First Name:	Enter_TKS	Parameter("First_Name")	
3	Last Name:	Enter_TKS	Parameter("Last_Name")	
4	Meal:	Select_TKS	Parameter("Meal")	
5	Card Type:	Select_TKS	Parameter("Card_Type")	
6	Number:	Enter_TKS	Parameter("Number")	
7	Expiration:	Select_TKS	Parameter("Expiration")	
8	Expiration:_2	Select_TKS	Parameter("Expiration_2")	
9	First Name:_2	Enter_TKS	Parameter("First_Name_2")	

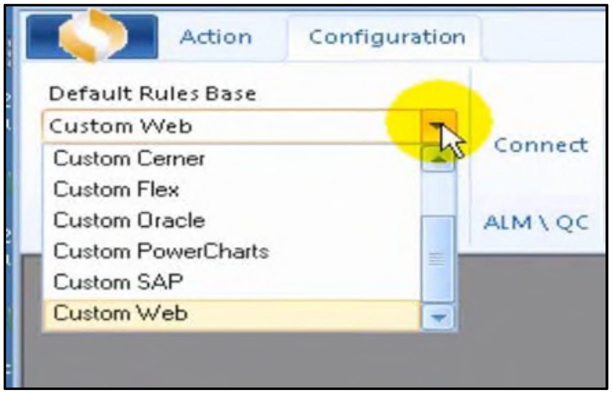
Figure 16: Screenshot from cFactory demo by Stephen Rosing in 2013[123]

This screenshot is a window that belongs to cFactory. In it you can see the components that have been created based on the objects that the UFT scanner has discovered. You can see that they have been parameterized. Since the user didn't manually do it, this is obviously auto-parameterization.

[redacted].[124] A rules base is a set of rules for interpreting a set of objects on a screen, paired with a method for interpreting those rules, in order to create a set of business objects that relate to those objects in some "smart" way. Rules bases create a flexible and extensible way to identify higher level business functionality by examining lower level objects, and they provide that capability across many technologies.

The rules bases are complex. TurnKey personnel have not disclosed exactly how large they are, but I would estimate that each one must be thousands of lines of code in order to do what they need to do. A different rules base is required for every different technology.

But their *existence* and use is not secret at all. The TurnKey demo video discusses that, and tells us what they are and how to select them.



Default Rules Base: Custom Web

Figure 17: Screenshot from cFactory demo by Stephen Rosing in 2013[125]

[123] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

[124] Deposition Ryan Jacques (June 8, 2016), at 83.

[125] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

> My application maybe extremely customized, and my test requirements are unique as far as I'm concerned. How does cFactory know what to create? **What we have is what's called a rules base. This little database cylinder or icon that we see in the corner. What we do is we provide a prebuilt list of rules for automating web Java, .net, SAP, and so on, different applications, but these rules are completely configurable so that you can say, 'OK, we're using this particular widget from this vendor,'** and we've built it into our application….
>
> Remember what I showed you. First was how do we create components? By clicking on the screen. We saw the concept with PowerPoint. Let me show you the live app. **This is cFactory and these are the rules bases. Notice Web, Cerner, Flex, Oracle. These are those rules bases that I told you about that we can say, we're going to test a Web app, or a Flex app, or an Oracle app, and that I can configure these rules to say, 'Oh, when you run across an Oracle data grid, I want to do this, this and this.' When we're testing my Web application and we see these particular widgets that I put into it, here's what I want you to do. We're going to go into this application.** We're going to say we're working with web. Then I click new component. Here's our ever faithful Tours app. I click on it. It says, 'Is this the screen you meant to click?' Yes. 'Do you want me to parse it?' Yes. Here's the components I can create. Pick the one you want. There it is. That easy.[126]

This demo from 2013—available on YouTube as of this writing for nearly 38 months and viewed more than 5,000 times by members of the public—shows the details of cFactory's user interface and workings.

Also, the keywords user guides, such as the TKS Web Keywords User Guide (see Figure 18), are posted publicly on TurnKey's website and describe the keywords and parameters in detail, giving examples of the parameterization used:

[126] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YouTube (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

[redacted]

Figure 18: Excerpt from TurnKey Web Keywords User Guide[127]

Every user of cFactory not only *gets* to learn what the keywords are and how they work, but *must* learn this in order to use the product effectively. If this were kept secret from the users, cFactory would have little value. It is therefore not surprising that this material is included in the five days of training that each TurnKey customer receives.[128]

There are also marketing materials available online that mention TurnKey's rules base:

> As an application aware system, cFactory™ utilizes a **rules file** which describes how the application forms or screens to be tested are laid out and organized**. These rules files are packaged in the open-standards language XQuery, with rules files available for many of the major packaged applications used today**, such as Oracle E-Business Suite, PeopleSoft, Siebel, JD Edwards, SAP, and others. In addition, **custom rules files can be**

[127] *TKS Web Keywords User Guide* (2016), at 85.

[128] *See* Appendix B, at B16-18.

> **created** for any in-house or third party application supported by the HP Software platform, often in the matter of just a few days.[129]

> cFactory utilizes **pluggable rules-based technology**, allowing it to support not only most major packaged applications such as ERP and CRM systems, but in-house developed and 3rd party-sourced applications as well. **Rules files can be readily customized** to match specific application UI layouts and technologies, allowing cFactory to support all application environments supported by HP's Unified Functional Testing toolset.[130]

And the cFactory User Guide[131] tells you how to edit the rules base:

[redacted]

Figure 19: Rules base format and inner workings explained in the cFactory User Guide[132]

What we see is that the existence and structure of the rules base, which is the so-called "intelligence" behind TurnKey's approach to auto-parameterization, is not secret in any way. The behavior of the rules base and resulting parameterization is also not secret. The only thing that seems to be secret is the source code for the rules bases.[133]

This Alleged Secret Was Already Practiced By HPE

Mr. Ellis claims:

> BPT v. 12.5x has numerous important similarities to the cFactory solution and the confidential methodologies TurnKey disclosed to HP in October 2014. First and most basic is the fact that it performs autoparameterization.[134]

[129] *TurnKey cFactory Solution Brief*, ORASI (2009), at 2, *previously available at* www.orasi.com/documents/TurnKey-cFactory-Solution-Brief.pdf.
[130] HP00041642.
[131] *TurnKey cFactory 2.6 User Guide: Revision 1*, at 163-166
[132] *TurnKey cFactory 2.6 User Guide: Revision 1*, at 162.
[133] …and that *remains* secret. HPE was not shown that or given that, and does not possess it.
[134] Ellis Disclosure, at 15.

Based on Turnkey's code, documentation, and depositions, it is my understanding that a "rules base," as that term is used by TurnKey, is a file that contains instructions on how to analyze test objects and decide how to create business components that represent them. cFactory rules bases are written in a rare language called [redacted]. I am told that no one on the HPE development team uses [redacted]. It is not used in any HPE testing product.

There appears to be one very important element that is starkly different between cFactory and BPT auto-parameterization technologies: *cFactory uses a rules base which contains executable code, and BPT does not*. Even though BPT does have an XML file that contains instructions in the form of a list of object types to exclude from auto-parameterization, there is no code in it. This kind of rules base is a reference file but not an independent program that executes, as is the case with cFactory. This fact alone makes clear why HPE would not want to misappropriate TurnKey's methodology: it would be far too expensive to copy TurnKey's approach. It would require a whole new layer of software and interfaces.

Using a rules base as TurnKey does requires that a lot of code be written that is specific to each and every different technology that might be found in an application-under-test. It is a major undertaking to create a rules base, and more work still to maintain it.

Based on my discussions with the HPE team, I know that HPE does not make use of cFactory-style rules bases. Instead it uses a simple method of identifying components and parameters when learning a flow for an SAP GUI product: BPT simply asks SAP for that information. This is why one of the auto-parameterization and component generation features that TurnKey is complaining about in BPT Version 12.5 (the SmartRecord feature) only works for SAP applications. Other technologies do not necessarily have what's called an "application map" that provides the metadata that BPT needs to separate out the components and parameters automatically.

Analysis Of Secret #2: Auto-Generation Of A Data Repository

This Alleged Secret Was Not Specified With Reasonable Particularity

This is what TurnKey provides as the definitive description of its Auto-Generation of a Data Repository secret:

> [redacted]

> [redacted][159]

At first glance, especially to the non-technical eye, this description may seem specific and thorough. It is in fact vague and confusing. This is not a description of an algorithm, but rather a loosely connected list of partially described behaviors. For instance:

- **"Test component" is not defined.** Does TurnKey mean to limit this secret to HPE-style keyword-based business components? This is important because although a general-purpose test script is a test component, TurnKey elsewhere asserts that their secrets pertain only to "scriptless" technology.

- **No logical connection is made between generating components and creating the data repository.** And indeed in cFactory's own product there is no connection. In TurnKey's technology, the product that creates the data sheets is called the Data Sheet Creator, is a separate tool and process from that which generates the components.

- **It does not actually specify generating a data sheet.** In this description, the first sentence presents only a high-level statement that a data sheet is created and what it looks like. There is no discussion of *how* that data sheet is created. Created by what? By whom? At what time? The rest of the description presumes that the data sheet has already been created.

- **It fails to explain the difference between "exporting at least one parameter to the data sheet" and "updating at least one column heading associated with at least one parameter."** I believe they mean exactly the same thing, but neither of them are specified any further.

- **The description does not match its topic.** Most of the description pertains to change detection and handling rather than auto-generating the data repository.

Nowhere in this alleged secret are there instructions about how to implement this idea. Nowhere is the inputs or the output specified. It is a vacuous description. It teaches nothing.

One aspect of this alleged secret that Mr. Ellis specifically discussed in his report is the naming of columns in the data sheet:

> This formatting of the Microsoft Excel sheets parallels the TurnKey methodology, in which the labels on the columns in the datasheets are set to match the names of the UI controls on the application screens for which they hold testing data. This is enabled by having the component parameters automatically named to match the names of the UI controls on the application screen. As parameters are created, the parameters are given the same name as the UI control on the application screen for the component step to which they are tied.[160]

To demonstrate the general problem with all of TurnKey's attempts to describe its secrets with particularity, I have scoured the testimony of Mr. Ellis and Mr. Kallaway, looking for details behind the alleged secret of the naming of columns. Here is everything I could find, along with my own paraphrasing of their testimony:

[159] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).
[160] Ellis Disclosure, at 21.

Kallaway Deposition[161]

		Testimony	My Paraphrase

[redacted]

[161] Kallaway Deposition (May 17, 2016).

Kallaway Deposition[161]

		Testimony	My Paraphrase

[redacted]

Ellis 30(b)(6) Deposition[162]

		Testimony	My Paraphrase

[redacted]

[162] Ellis 30(b)(6) Deposition (Feb. 24, 2016).

Ellis 30(b)(6) Deposition[162]			
		Testimony	My Paraphrase
[redacted]	[redacted]	[redacted]	[redacted]

Ellis 30(b)(6) Deposition[162]

		Testimony	My Paraphrase
[redacted]	[redacted]	[redacted]	[redacted]

Taking all of the paraphrased statements from above, I can construct a concise statement of the alleged secret corresponding to TurnKey's allegedly confidential and proprietary method for naming column headers in cFactory:

[redacted]

Based on this analysis, I conclude that the almost everything TurnKey has actually specified is cFactory behavior already in plain view, i.e. the fact that an Excel spreadsheet appears, with columns in it that are named after elements on the screen of the application-under-test. The only things not in plain view, and that aren't also references to HPE technology, are items 6 and 15, in the list above: the rules base, and the fact that "some properties are stored." But neither of those things is specified to any level of detail, much less a reasonable level of detail. This is not any sort of viable algorithm that would allow me to understand what the product does behind the scenes. Instead it is a statement of what a user might *want* a product to do, couched in a language that users of these products would readily understand. In other words, this is marketing speak, not a technical blueprint.

Every instance where I have highlighted text in orange is an example of something important that was left out of Mr. Ellis' and Mr. Kallaway's descriptions of column naming. Each of those instances is where HPE, if it had wanted to implement TurnKey's methodology, would have had to ask "yes, but *how* do you do this?"

Thus, if there is a real secret here, it has not yet been described with reasonable particularity.

This Alleged Secret Derives No Value From Secrecy

There would be no point for TurnKey to keep secret the disclosures I've analyzed above regarding automatic generation of a data repository. In fact, if they didn't talk about features such as the fifteen-point list, above, customers would think cFactory was not a state-of-the-art tool. Maintaining a test data repository is essential for tools of its type.

TurnKey may argue that it is not talking about the general concept of creating a test repository, creating columns in that repository, and associating columns with data in a test, etc., but rather some specific *methodology* for doing that. As I argued above, if it has such a specific methodology then it has not been revealed through any testimony, filing, or written discovery response by TurnKey. What *has* been described is a general concept, and that concept is straightforward and widely implemented around the industry.

This Alleged Secret Was Already Known To The Industry

Auto-generation of repositories for test data is no secret. It is not a secret, in general; it is not a secret in each of the elements described by TurnKey; and it is not a secret when all of the elements described by TurnKey are considered together. There is absolutely nothing esoteric about using spreadsheets to store and access test data.

Test data is routinely stored separately from test scripts. Dedicated data repositories are a longstanding practice. Test data may be stored in files, databases, or specifically spreadsheets. Facilities to access the data automatically are pervasive and commonplace. Creating repositories automatically is also commonplace, going back to the dawn of computing.

Another publicly marketed test tool with similar functionality is Rational TestFactory, which employed a "test datastore" as early as 2001. The "test datastore" automatically saved data about objects in the applications it tested.

Here is a modern example: SmartBear's TestComplete. It creates test data and puts it into repositories. It has not kept it secret, either. I visited the SmartBear booth at the StarEast conference this spring and saw a demo of this feature. The TurnKey booth was a scant few paces away, at the time.

SmartBear's website explains:

> Note that you can create and populate such storages manually, or you can command TestComplete to do this automatically. For this purpose, TestComplete includes the Data Generator that lets you generate test data of various types and store it either to a TestComplete Table variable or to an Excel file. After you specify all the desired parameters for the generated data, TestComplete will automatically generate it according to the defined parameter values and save these values to the variable or file. This approach decreases the time spent on preparing data for data-driven tests. For more information, see the Using Data Generators section in TestComplete's help.[163]

The feature that SmartBear is referring to here was introduced into its TestComplete product in June 2014, and information about this feature would have been available to any industry observer who attends conferences or reads industry literature. SmartBear is a well-known company. If any of TurnKey's personnel looked at what competitors were doing at the time, they would have known about these capabilities in SmartBear's product.

This Alleged Secret Was Not Kept Secret By TurnKey

The data repository generation features of cFactory are in plain view—not just to any user, but to any member of the public. Here is another screen shot from the TurnKey's 2013 YouTube demo:

[163] *Introduction to Data Driven Testing*, SMARTBEAR (last visited Feb. 2016), https://smartbear.com/learn/automated-testing/introduction-to-data-driven-testing/.

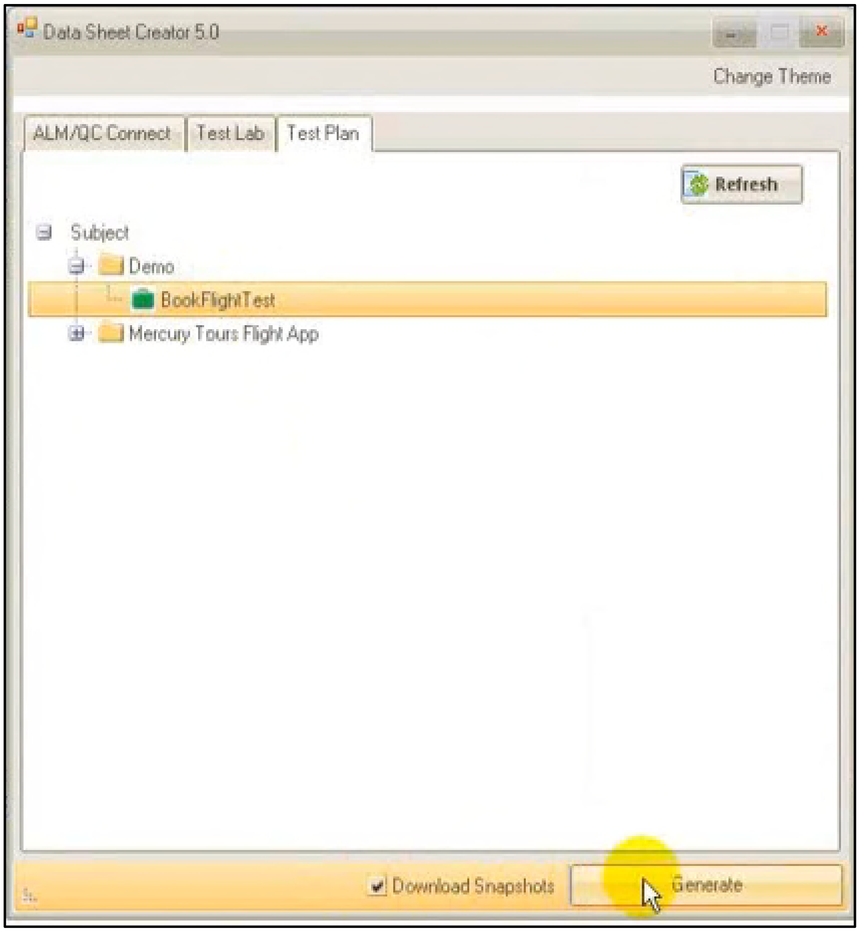


Figure 31: Screenshot of Data Sheet Creator as demonstrated publicly by Stephen Rosing in 2013[164]

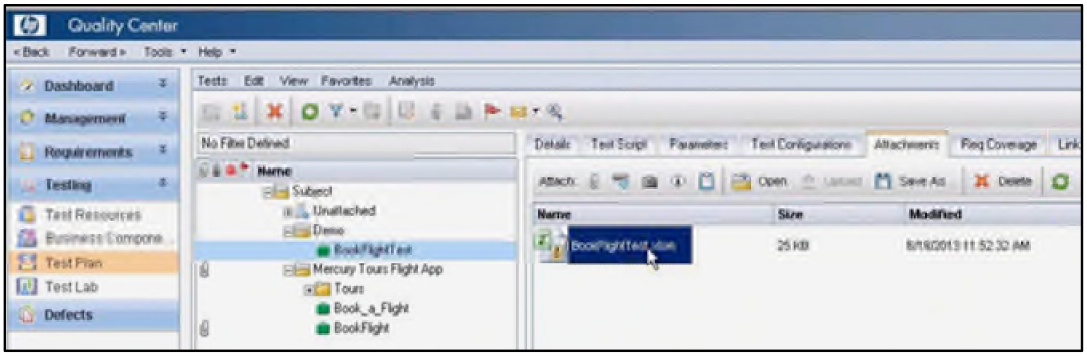
The first clue is the name "Data Sheet Creator 5.0." Also note that the button says "generate." Together, they tell you that a data sheet for your test is about to be generated automatically. A connection to HP ALM has been made so that the user can select the "BookFlightTest" test to make a data sheet.

In case that is not obvious enough, the narrator of the video explains:

> What do I need now? I need my data. How does this Excel spreadsheet get created? It's stored here under the attachments tab. To create that Excel spreadsheet, we provide a utility -- it's part of the solution, it doesn't cost extra -- called the Data Sheet Creator. As you can see here, I'm simply doing a remote login into this Quality Center project. There's my test, the book flight test. I click it and say, "Generate my Excel spreadsheet." It takes a few seconds. Hit "Refresh." There's the Excel spreadsheet.[165]

The following screenshot from the demo video shows the newly generated data repository that results from the above instructions.

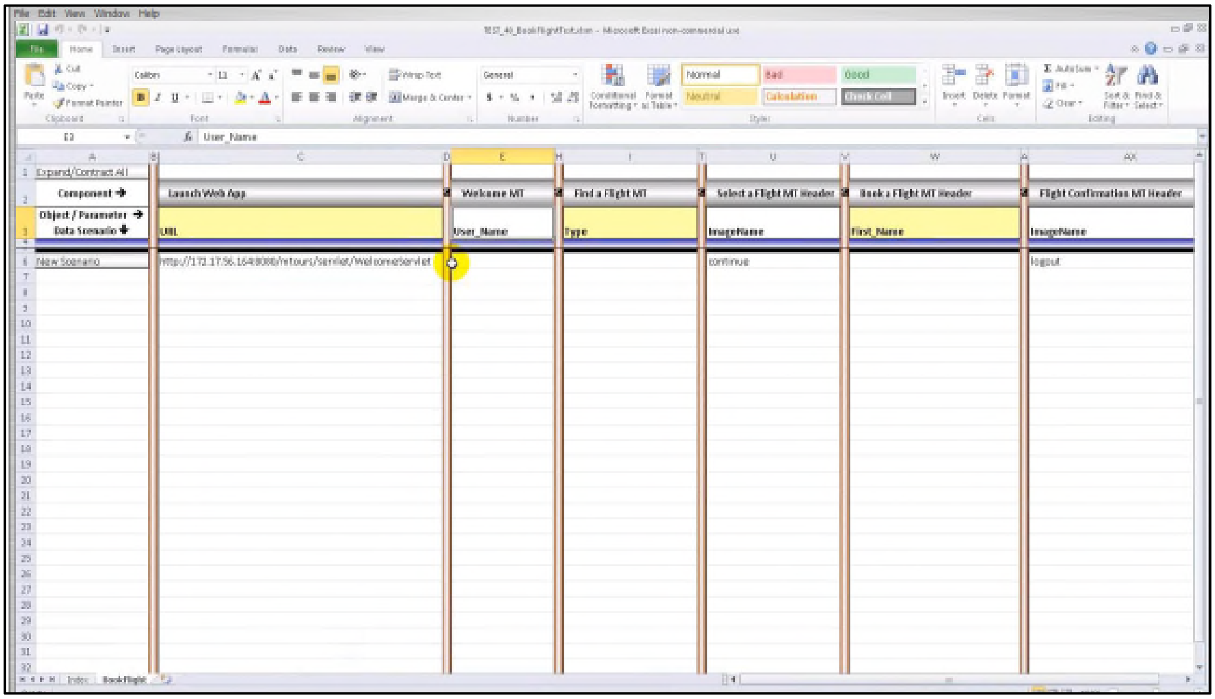
[164] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.
[165] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.



Name	Size	Modified
BookFlightTest.xlsm	25 KB	8/19/2013 11:52:32 AM

Figure 32: Freshly generated data repository spreadsheet as demonstrated publicly by Stephen Rosing in 2013[166]

There it is. The user clicks a button to create the data repository, and it gets created. But are its contents encrypted? Hidden? Secret? No. Actually, the cFactory *requires* testers to use this spreadsheet to edit their tests.



	A	C	E	I	U	W	AX
1	Expand/Contract All						
2	Component →	Launch Web App	Welcome MT	Find a Flight MT	Select a Flight MT Header	Book a Flight MT Header	Flight Confirmation MT Header
3	Object / Parameter → Data Scenario ↓	URL	User_Name	Type	ImageName	First_Name	ImageName
6	New Scenario	http://172.17.56.164:8080/mtours/servlet/WelcomeServlet			continue		logout

Figure 33: Spreadsheet generated by TurnKey software with columns named the same as test parameters[167]

In the video demo, the narrator tells us all about it:

> Let's open it up and take a look at it. **Here are those columns that I was talking about. Notice, there's one for each component.** If I click this little icon here, I can even see the screenshot. I say, 'Oh, this is the login screen.' If I double click that component, **I can see the**

[166] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.
[167] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

> **fields that I need to fill in, username, password, and the sign in button that I want to click.**[168]

Additionally, TurnKey's own marketing documentation talks about this:

> Using cFactory's™ TDMS™ data sheet generator, the analyst starts by selecting the desired test case or integration test. With a single mouse click, the generator analyzes all of the selected test cases including the screens touched, the regions and objects on those screens, whether the fields are optional or are required to be filled in by the application, etc. The TDMS™ generator then **creates a complete MS Excel workbook specifically designed and color coded to hold data for that integration test**.[169]

> The cFactory platform also includes automated testing's first data-driven, test data management engine, dataDriver. dataDriver is based upon the power and simplicity of Microsoft Excel, **allowing users to simply enter the testing data they want used into familiar spreadsheets created for them by the system**.[170]

> Once an end-to-end test flow is laid out, cFactory dataDriver test data management engine analyzes the flow and **creates a repository in Microsoft Excel, which will store, manage, and maintain the flow's testing data**. The testing data itself can be populated into the Microsoft Excel workbook in one of several ways. Data can be pasted or typed into the repository, test data can be automatically generated using dataDriver's embedded modules, and data can also be pulled directly from an application's metadata through direct database links made by specialized dataDriver modules.[171]

> Once we define the screen flow when creating our tests, the components will automatically be analyzed and a column for every required input and output necessary to drive the test will be generated in a formatted Excel workbook. This is Excel workbook is where we will identify the data we will use to drive the application through the processes we want to test. You will notice that there is a column for each component. These columns can be expanded to show each of the fields/objects on the application screen requiring data input, along with the controls. You simply input the data as you would on the screen.[172]

The following slide is from the presentations used during the October Briefings, which TurnKey has testified are not confidential[173]:

[168] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YouTube (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.

[169] *TurnKey cFactory Solution Brief*, Orasi (2009), at 2, *previously available at* www.orasi.com/documents/TurnKey-cFactory-Solution-Brief.pdf..

[170] HP00041642.

[171] HP00041644.

[172] *Turnkey Sales Presentation Script* (July 2014), *available at* http://turnkeysolutions.com/wp-content/uploads/2015/05/TurnKey-Sales-Presentation-Script.pdf.

[173] Kallaway Deposition (May 17, 2016), at 99.

[redacted]

Figure 34: "cFactory - Data Drive Architecture" slide from October Briefings[174]

Not only is this slide claiming that a data repository (here, an Excel workbook) is created automatically, it also highlights that the column name is the same as the component name, which TurnKey identifies as part of its "secret."

I also found a confidential sales demo script on TurnKey's website. In this script, the data generation features of cFactory are openly presented to potential customers:

> Once we define the screen flow when creating our tests, the components will automatically be analyzed and a column for every required input and output necessary to drive the test will be generated in a formatted Excel workbook. This is excel workbook is where we will identify the data we will use to drive the application through the processes we want to test.[175]

Finally, of course, the cFactory User Guide—which under the OEM agreement is declared not to be confidential[176]—explains auto-generation of a data repository in detail. *There's a whole chapter on it.*[177]

So, given these disclosures, and the fact that the feature itself is by its nature open and public, let me go back to TurnKey's actual words. This is exactly how it describes the secret:

> [redacted][178]

> [redacted]

[174] TK003822.

[175] *TurnKey Sales Presentation Script* (July 2014), *available at* http://turnkeysolutions.com/wp-content/uploads/2015/05/TurnKey-Sales-Presentation-Script.pdf. It's marked confidential. However it's publicly viewable on TurnKey's website.

[176] OEM Agreement §§ 1.3, 10.1 ("[redacted]").

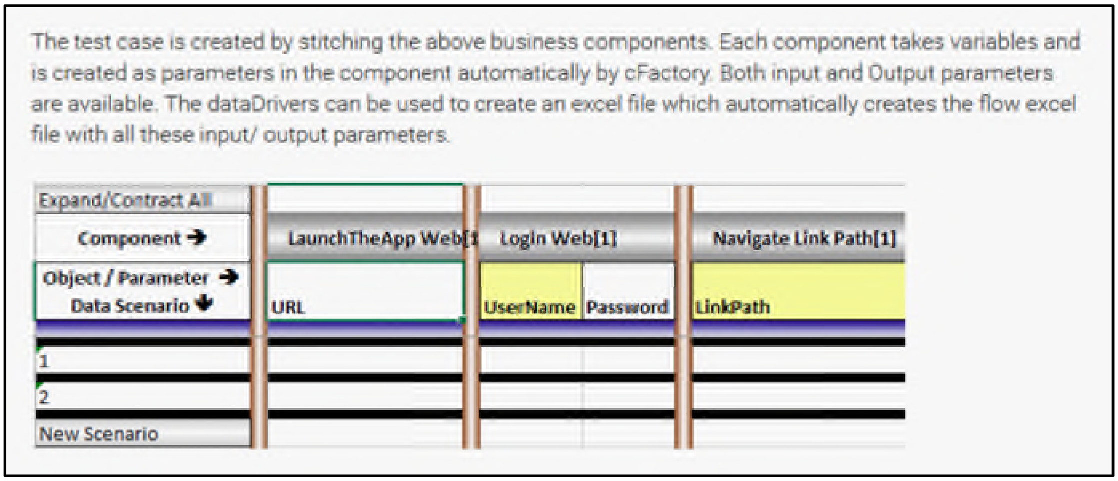
[177] TurnKey cFactory 2.6 User Guide: Revision 1 at 82-86.

[178] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).

[redacted][179]

Not a single word of either of those statements represents anything secret about cFactory.

People outside of TurnKey seem to know about cFactory and how it writes to Excel files, too. Writing for the "IT Convergence" blog, Rajasekhar Nanduru, Sai Prabha, and Manasa Mothe reviewed cFactory and described its features, including how it "automatically creates the flow Excel file with all these input/output parameters":



The test case is created by stitching the above business components. Each component takes variables and is created as parameters in the component automatically by cFactory. Both input and Output parameters are available. The dataDrivers can be used to create an excel file which automatically creates the flow excel file with all these input/ output parameters.

Figure 35: Blog review of cFactory published by third-parties[180]

This Alleged Secret Was Already Practiced By HPE

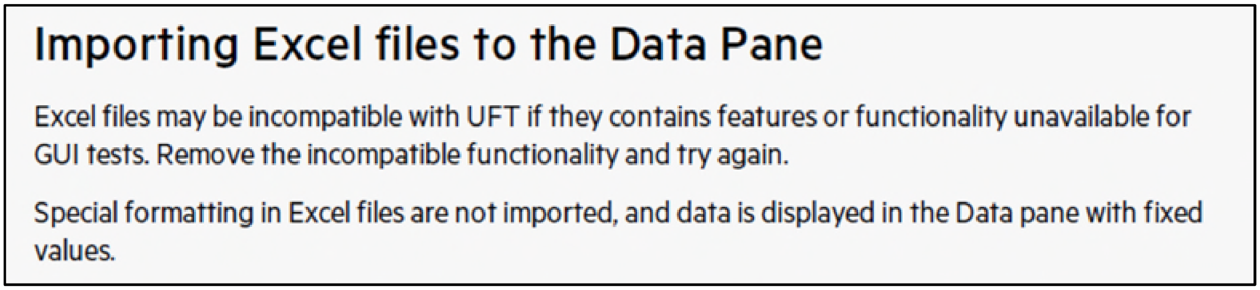
The claim that associating test components with test data is a TurnKey secret seems to be wrong on its face, for the simple reason that HP ALM already *is* a repository for test data that *has been automatically created* (during the installation process). It's entire purpose is to associate various kinds of data together in various ways. Its services as a data repository are extensively used by cFactory itself. TurnKey cannot deny this.

In case there was some important distinction or process described by TurnKey that I hadn't accounted for, I took a closer look at the details TurnKey offered about this secret in the Ellis Disclosure. The left column in the table below contains sentences from that report, with information about HPE's corresponding functionality on the right:

Assertions From Mr. Ellis' Expert Report[181]	**Capabilities of HPE's Products Before ALM/BPT/UFT Version 12.5**
"TurnKey's dataDriver technology and methodology utilizes Microsoft Excel as its primary data repository mechanism."	**Excel as Repository.** The HPE products have *long* facilitated using Excel as a data repository mechanism.
"This methodology provides users with an intuitive user interface that requires a minimum amount of	**Provide Spreadsheet Interface.** The HPE products have *long* had a spreadsheet interface built into

[179] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 54.
[180] *What is cFactory and How to Use it in Test Automation*, IT CONVERGENCE (Feb. 1, 2016), *available at* http://blog.itconvergence.com/what-is-cfactory-and-how-to-use-it-in-test-automation.
[181] Ellis Disclosure, at 19-20.

the Data Sheet Creator generates are nothing like the sheets that HPE generates. Although they are both Excel spreadsheets, the similarity goes no further. TurnKey's data sheets are full of macros and special formatting, which are completely missing from the spreadsheets that HPE generates. In fact, HPE documentation advises users to avoid any fancy formatting when working with Excel sheets:



Importing Excel files to the Data Pane

Excel files may be incompatible with UFT if they contains features or functionality unavailable for GUI tests. Remove the incompatible functionality and try again.

Special formatting in Excel files are not imported, and data is displayed in the Data pane with fixed values.

Figure 38: Excerpt from HP UFT Version 12.54 User Guide[189]

Analysis Of Secret #3: Automatic Detection Of Changes In A Business Application And The Updating Of BPT Components

This Alleged Secret Was Not Specified With Reasonable Particularity

The alleged secret is described in TurnKey's Response to Interrogatory No. 1 as follows:

> [redacted][190]

This is a precise and lawyerly way of saying "We scan the new version of the product to get the components, and then we compare those new components with the old ones and show the differences." Which in the technical world we would simply describe as "Scan again and do a diff."

No specific process of diffing is described here, and, again, no algorithm is provided. Is there anything special that would make this feature anything other than an ordinary diff process? Not according to TurnKey's Rule 30(b)(6) testimony:

> [redacted]

[189] *HP Unified Functional Testing User Guide, v12.54* (Sept. 2016), at 83.

[190] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).

> [redacted][191]

> [redacted][192]

In Appendix A[193], I analyze Mr. Kallaway's statements. Based on that analysis, I find them to be similarly lacking in specificity. If the descriptions Mr. Kallaway gave in his deposition are consistent with what he explained to HPE's employees during the October Briefings, then he did not explain this "secret" in *any* level of detail.

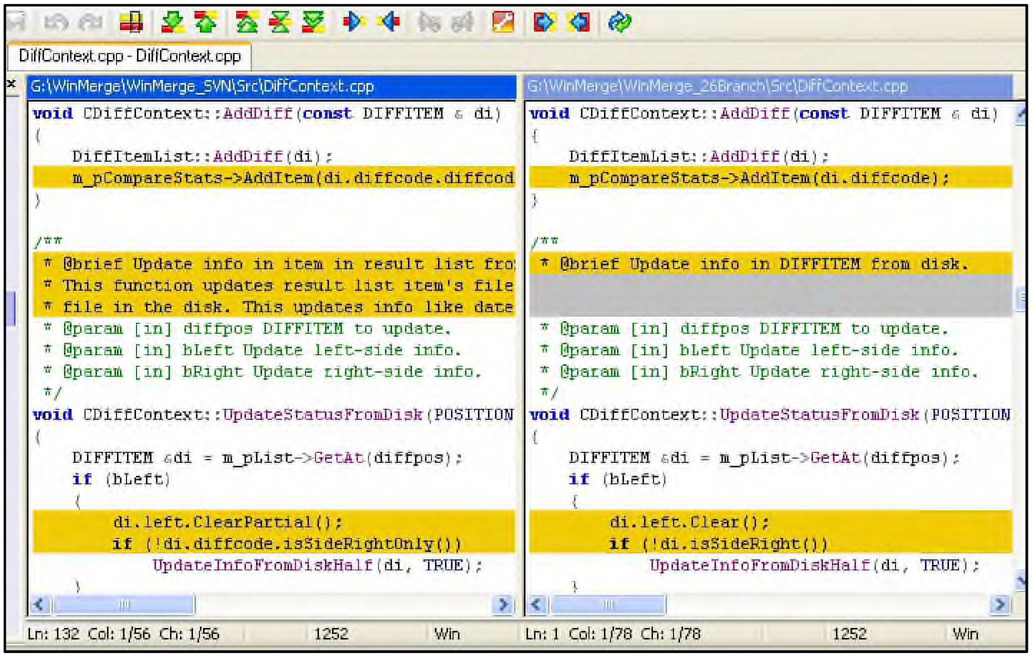
It is possible that TurnKey is trying to say that there is some sort of technological difference between comparing business components and comparing other kinds of things. If indeed that is their claim, then they have offered no evidence to support it. I don't see anything unique or difficult about comparing business components, and even if there were something difficult, and that something were secret, then it could not have been described with reasonable particularity by TurnKey—this is because I have carefully analyzed every word of testimony, every discovery response, and every filing TurnKey has made, and I have been unable to find *any* information I can recognize as other than vague or commonplace.

This Alleged Secret Was Already Known To The Industry

As I glean from the various filings and testimony, TurnKey claims that automatic detection of changes in a business application and the updating of BPT components is the process of identifying changes in a business application under test that affect BPT components, and providing a facility for user review and acceptance of those changes. This process requires the application learning capabilities of HPE UFT, the component creation capabilities of HPE BPT, and the test asset repository of HPE ALM.

This is not a secret. Change detection in data is a relatively simple matter. Visual comparison of data sets is a little more complicated, but still well established in the art. So-called "diffing" tools for comparing unstructured text data and presenting visual displays of those differences are old hat. The one I use most often is WinMerge (see Figure 39), an open source diff and merge tool that first shipped in 2000:

[191] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 87-88.
[192] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 91-92.
[193] *See* Appendix A, at A30.



G:\WinMerge\WinMerge_SVN\Src\DiffContext.cpp

```
void CDiffContext::AddDiff(const DIFFITEM & di)
{
    DiffItemList::AddDiff(di);
    m_pCompareStats->AddItem(di.diffcode.diffcod
}

/**
 * @brief Update info in item in result list fro
 * This function updates result list item's file
 * file in the disk. This updates info like date
 * @param [in] diffpos DIFFITEM to update.
 * @param [in] bLeft Update left-side info.
 * @param [in] bRight Update right-side info.
 */
void CDiffContext::UpdateStatusFromDisk(POSITION
{
    DIFFITEM &di = m_pList->GetAt(diffpos);
    if (bLeft)
    {
        di.left.ClearPartial();
        if (!di.diffcode.isSideRightOnly())
            UpdateInfoFromDiskHalf(di, TRUE);
    }
```

Ln: 132 Col: 1/56 Ch: 1/56 1252 Win

G:\WinMerge\WinMerge_26Branch\Src\DiffContext.cpp

```
void CDiffContext::AddDiff(const DIFFITEM & di)
{
    DiffItemList::AddDiff(di);
    m_pCompareStats->AddItem(di.diffcode);
}

/**
 * @brief Update info in DIFFITEM from disk.

 * @param [in] diffpos DIFFITEM to update.
 * @param [in] bLeft Update left-side info.
 * @param [in] bRight Update right-side info.
 */
void CDiffContext::UpdateStatusFromDisk(POSITION
{
    DIFFITEM &di = m_pList->GetAt(diffpos);
    if (bLeft)
    {
        di.left.Clear();
        if (!di.isSideRight())
            UpdateInfoFromDiskHalf(di, TRUE);
    }
```

Ln: 1 Col: 1/78 Ch: 1/78 1252 Win

Figure 39: WinMerge screenshot[194]

Automatic detection of changes with the ability to *choose* the changes you want to accept is yet another feature that industry already knows about. Every programmer who works on a development team is familiar with diffing tools.

It is also, according to Ryan Jacques of TurnKey, an obvious feature to have added:

> [redacted][195]

This Alleged Secret Derives No Value From Secrecy

This is another instance where, if TurnKey really did keep secret the process of change detection, they would have a lot of trouble selling their tool. TurnKey considers this an important feature in cFactory—a major selling point of TurnKey.

[194] *WinMerge Alternatives*, TECHNOLOGYSPHINX (Feb. 17, 2016), *available at* http://www.technologysphinx.com/winmerge-alternatives/.

[195] Jacques Deposition (June 8, 2016), at 94-95.

This Alleged Secret Was Not Kept Secret By TurnKey

In addition to being already known within the field, this feature is, once again, self-revealing; it sits in plain sight. It is widely advertised, demoed, and otherwise readily apparent to any user or potential user. If it were ever supposed to be a secret, TurnKey has not treated it that way.

Change detection is discussed in the sales slide deck used in the October 2014 briefings. This slide deck was marked confidential, but TurnKey has testified that it contains no secrets:[196]

[redacted]

Figure 40: "cFactory - Auto-maintenance View" slide on display[197]

Change detection also features prominently is Stephen Rosing's 2013 demo video on behalf of TurnKey. He provides a description just as detailed as that given by Mr. Kallaway in deposition:[198]

> Last thing I wanted to show you is the maintenance. We talked about, when a screen changes that we're going to show you how it affects the test. Let me pause here for just one second and tell you what's going on. This application here...I showed you Oracle earlier. Now here's SAP. This is a screen that I have created a component for, and I've been using it in a test. Development just told me, 'Hey, we just applied a patch, and we changed this screen.'
>
> Here's what I'm going to do. I'm going to go into cFactory, and I'm going to open my old component. I'm going to go in and open up that component, which is Functional Location Structure. I just called it 'Old' so that it was easy to see. There it is. Now I'm going to click update, and I'm going to click the new screen. We see the same Learn screens as we saw before. Only now, when it parses the screen, it's going to return a Differences Viewer....[199]

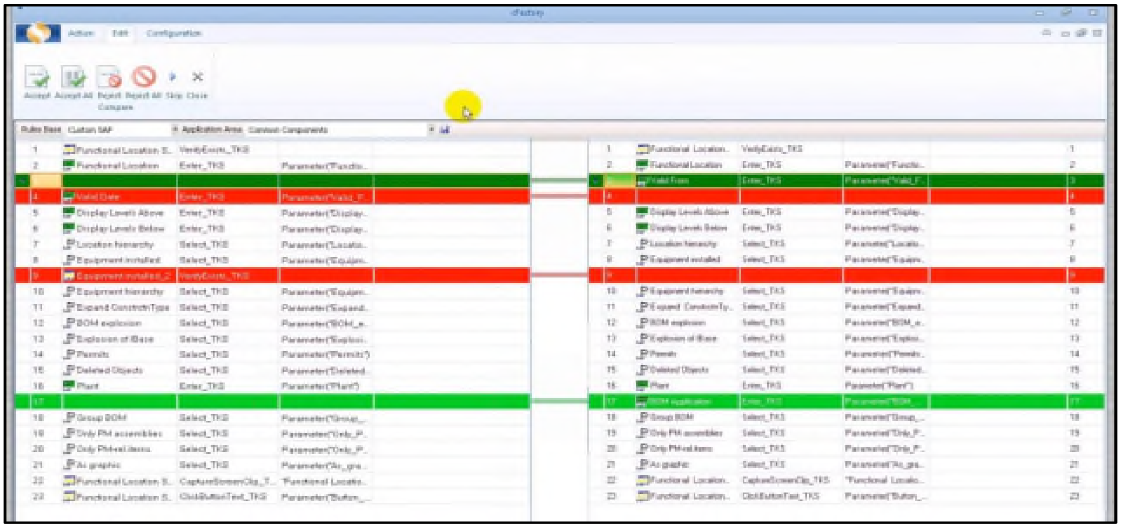
The following screenshot shows the screen displayed during this part of Mr. Rosing's demonstration:

[196] Kallaway Deposition (May 17, 2016), at 99.

[197] TK003825.

[198] Appendix A, p. A30

[199] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YouTube (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.



Rules Base: Custom SAP
Application Area: Common Components

1	Functional Location S...	VerifyExists_TKS	
2	Functional Location	Enter_TKS	Parameter("Functio...
4	Valid Date	Enter_TKS	Parameter("Valid_F...
5	Display Levels Above	Enter_TKS	Parameter("Display...
6	Display Levels Below	Enter_TKS	Parameter("Display...
7	Location hierarchy	Select_TKS	Parameter("Locatio...
8	Equipment installed	Select_TKS	Parameter("Equipm...
9	Equipment installed_2	VerifyExists_TKS	
10	Equipment hierarchy	Select_TKS	Parameter("Equipm...
11	Expand ConstructnType	Select_TKS	Parameter("Expand...
12	BOM explosion	Select_TKS	Parameter("BOM_e...
13	Explosion of IBase	Select_TKS	Parameter("Explosi...
14	Permits	Select_TKS	Parameter("Permits")
15	Deleted Objects	Select_TKS	Parameter("Deleted...
16	Plant	Enter_TKS	Parameter("Plant")
17			
18	Group BOM	Select_TKS	Parameter("Group_...
19	Only PM assemblies	Select_TKS	Parameter("Only_P...
20	Only PM-rel.items	Select_TKS	Parameter("Only_P...
21	As graphic	Select_TKS	Parameter("As_gra...
22	Functional Location S...	CaptureScreenClip_T...	"Functional Locatio...
23	Functional Location S...	ClickButtonText_TKS	Parameter("Button_...

1	Functional Location...	VerifyExists_TKS		1
2	Functional Location	Enter_TKS	Parameter("Functio...	2
	Valid From	Enter_TKS	Parameter("Valid_F...	3
4				4
5	Display Levels Above	Enter_TKS	Parameter("Display...	5
6	Display Levels Below	Enter_TKS	Parameter("Display...	6
7	Location hierarchy	Select_TKS	Parameter("Locatio...	7
8	Equipment installed	Select_TKS	Parameter("Equipm...	8
9				9
10	Equipment hierarchy	Select_TKS	Parameter("Equipm...	10
11	Expand ConstructnTy...	Select_TKS	Parameter("Expand...	11
12	BOM explosion	Select_TKS	Parameter("BOM_e...	12
13	Explosion of IBase	Select_TKS	Parameter("Explosi...	13
14	Permits	Select_TKS	Parameter("Permits...	14
15	Deleted Objects	Select_TKS	Parameter("Deleted...	15
16	Plant	Enter_TKS	Parameter("Plant")	16
17	BOM Application	Enter_TKS	Parameter("BOM_...	17
18	Group BOM	Select_TKS	Parameter("Group_...	18
19	Only PM assemblies	Select_TKS	Parameter("Only_P...	19
20	Only PM-rel.items	Select_TKS	Parameter("Only_P...	20
21	As graphic	Select_TKS	Parameter("As_gra...	21
22	Functional Location...	CaptureScreenClip_TKS	"Functional Locatio...	22
23	Functional Location...	ClickButtonText_TKS	Parameter("Button_...	23

Figure 41: Maintenance mode screen as demoed publicly by Stephen Rosing in 2013[200]

Mr. Rosing continues:

> … If I open this up, I can see two controls have been added, two controls have been deleted. Good. Let's accept these changes. Close the Differences Viewer. There's my new component. I say, save it. Once I save it, the next thing cFactory is going to do is say, 'Hey, here's the tests that you have that use it.' For example, this one, supply chain. I say, update it. It says, 'Updated successfully.'
>
> Let's go find that supply chain test in Quality Center. There we go. Supply chain, open up that Excel spreadsheet. There's the color coding for the controls that were deleted. If we were to scroll to the left, we would see the green ones for the controls that were added.
>
> Now, I've updated my component, which updated the test that it's being used by. Now, I simply need to say, I don't need to provide this data in the next test run, but there are two fields that I do need to add now. It's literally that easy to maintain your test cases.[201]

Also, there is a chapter about change detection in the cFactory User Guide:

[redacted]

Figure 42: Chapter 7.1.1 on "cFactory Comparison Mode," from the cFactory User Guide[202]

[200] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.
[201] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.
[202] TurnKey cFactory 2.6 User Guide: Revision 1, at 126-147.

This text appears to be describing the "secret" cFactory uses in essentially the same terms that TurnKey uses in its Response to Interrogatory No. 1.

This Alleged Secret Was Already Practiced By HPE

Every aspect of this alleged secret has long been practiced by HPE. HPE had it before the October 2014 Briefings, and even long before the 2012 TurnKey/HPE OEM agreement was made.

As I wrote in my opening report, change detection is an essential feature of any test tool that performs user simulation. Testers must be able to detect whether a product is looking or behaving differently than it did when it was tested earlier. Testers may also need to compare different versions of their test scripts and objects. Although there are many different applications of change detection, and many different looks, they are all fundamentally the same technology. ALM, UFT and BPT offer several features for comparing an AUT with an earlier version of that application in order to detect changes made to the newer version.

TurnKey has complained that the Change Detection Mode (CDM) in Version 12.5x was created with the help of alleged trade secrets received from it. This *cannot* be true for at least the following reasons:

1. **Testimony Shows That BPTEE Technology Is The Basis For Key Features In v12.5x.** HPE personnel have testified that they already possessed this secret, in that CDM is "revived" code from the BPTEE technology base.

2. **Similar Functionality Existed Long Before October 2014.** In addition to Change Detection Mode, HPE technology includes several other change detection, review, and resolution schemes that prove change detection, in general, is a longstanding and commonplace practice which takes many forms.

3. **The Accused Functionality Existed Long Before October 2014.** The documentation for CDM in BPT 12.53 is essentially the same as corresponding documentation in five earlier versions of HPE products that are not accused of being the fruit of misappropriation, and that documentation publicly reveals each important element of TurnKey's alleged secret.

4. **Accused Functionality Is Disclosed In Patent Applications From 2008.** In 2008, HPE filed two patent applications that describe the same technology that TurnKey claims was misappropriated.

Testimony Shows That BPTEE Technology Is The Basis For Key Features In Version 12.5x

Mr. Wigelman testified that his team "revived" the BPTEE code and integrated it into Version 12.50:

> For 12.5, UFT 12.5, the usability was like -- we added -- or actually revived the functionality that was already existing in -- in the BPT for SAP module. And we used the same code that is part of the same features of UFT also for BPT.[203]

> Q. So the auto parameterization that exists in BPT for 12.50 and higher, do I understand your testimony correctly that some of that came from BPT for SAP, which is the same thing as BPTEE, and some of it came from UFT?

[203] Wigelman Deposition (June 2, 2016), at 15.

Analysis Of Secret #4: Automated Component Generation

This Alleged Secret Was Not Specified With Reasonable Particularity

TurnKey describes this secret as follows:

> [redacted].[216]

This process description is another vague wish list. It contains no specific computable steps, and no algorithm. If this secret had been read, verbatim, to the HPE team by Mr. Kallaway, there is nothing HPE could have done with it. I reviewed Mr. Ellis' Rule 30(b)(6) deposition testimony, as well as Mr. Kallaway's testimony[217]—neither sheds any more light on this topic. Based on what TurnKey has disclosed over the course of this litigation, this is the best I can do to paraphrase TurnKey's fourth alleged trade secret, including all the operational information that TurnKey actually provides:

TurnKey claims that automated component generation is the automated process of creating BPT components with appropriate steps and parameters within HPE ALM. This is a four-step process:

1. [redacted]

[redacted]

[redacted]

[redacted]

Steps 1 and 4 are actually performed by HPE, not a tool that TurnKey developed. There are only two things about this alleged "secret" that don't appear in the other alleged secrets: (1) *components are automatically created,* and (2) *a scanning process is used* to gather information about the objects from which those components are created.

[216] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).

[217] I analyzed Mr. Kallaway's attempt to explain this alleged secret on page A25 of Appendix A.

- **Creation of components:** TurnKey has never explained *how* to decide what components to create beyond the fact that there is a process of some kind that involves something called a rules base, and which uses HPE Business Process Testing APIs to configure and store the components within ALM.

- **Scanning:** The scanning process is performed entirely through HPE's tool: UFT. So, there is no need to explain how that works. HPE already knows how that works.

This Alleged Secret Was Already Known To The Industry

Automated component generation is not a secret.

Automated component generation as TurnKey describes it is a specific example of the general process of automatically creating data structures or assets to be used in test automation. Such automated creation is well established in the art, going back to the very first record-and-playback systems—automated generation is what the word "record" refers to.

Automated component generation is just another way of saying "automated script generation." Automated script generation is established technology. Tools from Worksoft, SmartBear, SAP, Tosca, and IBM all do this.

Scanning application screens is well known, too—no matter if we are talking about scanning the entire screen or just some region of it. The free and publicly available AccExplorer32 tool from Microsoft has had scanning capabilities since 1998. In 2001, Rational Test Factory used a scanning process to map applications for testing.

This Alleged Secret Derives No Value From Secrecy

Although a company could produce a specific algorithm for creating components that gives the company an advantage over a rival, that cannot be the situation here. TurnKey has specified no algorithm in connection with this alleged trade secret. It revealed no algorithm during the October Briefings, nor did it reveal any algorithm during the course of this litigation. All that TurnKey has disclosed is that components of some kind are created by some process; and that scanning is somehow involved. These are selling points. As such, they only have value if they are *not* secret.

This Alleged Secret Was Not Kept Secret By TurnKey

As I argued in the section about auto-parameterization, automatic component generation cannot be a secret because (1) it is self-revealing, and (2) TurnKey has not taken steps to keep it secret. The same information that TurnKey now claims as its trade secrets can be gleaned from the many YouTube videos, training videos, and other publicly available documentation on its website.

With respect to component generation, TurnKey has emphasized the idea of "one-click learning," i.e. scanning an application without the user having to specifically click on each field. This is not an innovation. In fact, TurnKey relies on HPE UFT to perform these functions, and UFT is simply doing what other products have done before it. For example, Microsoft AccExplorer32 has been doing this kind of scanning since 1998, for testing purposes, in a testing context.

One-click learning is advertised by TurnKey:

> Simply **walk through your application screens** as the software identifies and **creates a library of test components** for easy customization and reuse.[218]

> The software enables even non-technical business users to create tests by simply clicking through application screens. It then analyzes the screens and automatically builds reusable, data-driven testing components for customization and reuse.[219]

> Harnessing the power of 3rd Generation (3G) testing technology, cFactory™ analyzes your application forms or screens and creates fully automated, data-driven test components, without scripting or recording, automatically….
>
> Using its 3G engine, cFactory™ **analyzes the forms or screens of an application and builds keyword-driven automated testing components** for that application automatically….
>
> The first step in automating the testing of an application using cFactory™ is to **capture or learn the screens of the application** using a cFactory™ version of HP's QuickTest Professional (QTP) toolset. **The Analyst launches QTP's 'Add Object' Wizard and clicks on each of the application screens to be automated. This captures or 'learns' the screens, after which the Analyst moves to cFactory™ and clicks the 'Analyze' button at the bottom of the window. cFactory™ then analyzes all of the screens, including all of the objects such as fields and buttons that each one contains, etc.** In just seconds, cFactory™ will display all of the automated test components that it can create for the selected screens….
>
> cFactory™ automatically breaks individual screens into different regions, such as separate tabs, header or footer sections, etc., and **assigns a test component to each region**….
>
> The Analyst **selects which components to create, or chooses the 'All Components' option, and selects the Quality Center folder to place the finished test components into. The 'Build' button is then clicked, and the selected test components are created and placed into Quality Center automatically**. cFactory™ can build hundreds of test components covering dozens and dozens of screens in just a few minutes, allowing a single Analyst to create a large volume of automated test components quickly….[220]

This marketing text reflects the same level of detail as what TurnKey provided to HPE during the course of this litigation in it various descriptions of its alleged trade secrets, including with respect to automated component generation.

TurnKey's training videos[221] provide much more detail about how cFactory works. Here is a segment from one 26-minute video:

> We're going to just do a basic learn. Another video will cover the advanced learn option. For this one, we're just going to click on the learn button. You'll see QTP come up, and then disappear. As soon as your mouse turns to a pointing hand, it means that it's ready for learning for the screen.

[218] *Scriptless Test Automation for HP ALM Quality Application Delivery without Compromise*, TurnKey Solutions (2014), *previously available at* www.turnkeysolutions.com.

[219] *Scriptless Test Automation for HP ALM: cFactory*, Orasi (2014), *previously available at* www.orasi.com.

[220] *TurnKey cFactory Solution Brief*, Orasi (2009), at 2, *previously available at* www.orasi.com/documents/TurnKey-cFactory-Solution-Brief.pdf.

[221] These videos are intended apparently for its customers but are posted publicly on its website and bear no confidentiality notice.

> You see my mouse is now a pointing hand. I can click anywhere in my application. If, for some reason, you forgot to layer cFactory right on top of your application, you can hold your control key down. You'll notice the mouse turns back into a pointing hand. You can click on your application, and bring it back up. Then if you release the control key, your mouse turns back into a pointing hand, meaning it's ready to learn the screen.
>
> **Again, just click anywhere inside the application. You want to select the page object. The page object is a parent. If I select this one, it will capture all of the objects underneath that parent object. Then I'll click OK. You'll get a define object filter window. You want to choose the selected object types, and choose the object types that are designed for your application.**
>
> **Whether they're just web objects, or if TurnKey has developed custom extensibility for your web app, you would select the TKS objects. To select those, you click on the select button, and go through the list, and just check the ones that you want to learn in your application.**
>
> That should be in your manual, or your trainer should have sent them to you. Once you're done clicking your objects, you click OK. On this define object filter, you click the OK button to move forward. It's going to analyze the screen, and build an object repository.[222]

The text I bolded discusses “parent objects,” “object filters,” and “object types.” This language is at least as technical as that of Mr. Ellis and Mr. Kallaway. The video continues:

> Then it's going to bring cFactory back up, where you can select the type of component you want to build. For every time you learn a component, you're going to get this option. You may have more in this list. You may just have the two. It depends on the application screen.
>
> In this case, I just have the one region of the screen I'm learning, so I'm getting two options. You'll see it's Oracle iProcurement Shop, and Oracle iProcurement Shop verification. The title of my screen is Oracle iProcurement Shop. **That's where it's getting the name. It's the page name of the application screen I'm learning.**[223]

The text I bolded explains how the object naming works. HPE also gets object names from the screen.

> Then you'll see the component in the cFactory window. The thing to note is the very first step of every component that you will have is going to be this operation load data. This is what ties the component to your data sheets. You'll next see a verifyexists_TKS keyword. That's just saying, "OK, verify that I'm on the right screen," in this case, this Oracle iProcurement Shop page. It's verifying that it's on the right screen. Then it gets into the actual items, and has a step for each individual item on this screen.
>
> You'll notice that they're all action types of keywords. There's no verifications. That's because I built the entering component. For the list box item type, it's a dropdown. It's using the select_TKS keyword. Then it has a parameter with the same name as the object that will show up in the data sheet. That's where you actually provide the data to enter into that field.
>
> Then all the web edits have an enter_TKS keyword with a parameter that has the same name as the edit field. That's saying, "OK, now I want to enter data into here." I'm going to

[222] *Creating Web Components*, TURNKEY SOLUTIONS (last accessed Feb. 2017), *available at* http://turnkeysolutions.com/wp-content/uploads/2016/03/Creating-Web-Components-Video.wmv.
[223] *Creating Web Components*, TURNKEY SOLUTIONS (last accessed Feb. 2017), *available at* http://turnkeysolutions.com/wp-content/uploads/2016/03/Creating-Web-Components-Video.wmv

> manually type in the data. That's what the application would do as well, if you were to run the script.
>
> The data that you provide in the data sheet for this parameter will be entered into that field. Checkboxes have the select_TKS keyword. You tell it whether you want to select the checkbox or not. It's going to order the steps from left to right, top to bottom. If I look at the screen, item type, contract number, it's going to go left to right, top to bottom.
>
> Then at the very bottom of the component, you're going to see three steps for the web. The first one here is step number 19. After all the edit fields and checkboxes on the screen, you're going to see a capture screen clip step. This ties to the capture screen clip field in the test lab. If that field in the test lab is set to yes, every time you run, and it sees this capture screen clip step, it's going to capture a screen clip, and save that to your run results. You'll see this as the bottom of every component. After the capture screen clip, you'll see a click button text. That's saying after you've entered the data, most of the time, you want to click a button or do some sort of navigation afterwards.[224]

This section is an excerpt from a much longer explanation of what is incorporated into components when they are created. Just from this excerpt we learn:

- The specific keywords used in the components and what they do are "enter_TKS," "verifyexists_TKS," and "select_TKS"
- LoadData is the first operation in every cFactory component.
- A capture screen operation is included in every component.
- cFactory orders the steps left to right and top to bottom.
- Parameters have the same names as columns in the data sheets.

This training video alone is 26 minutes long. There are no slides, and no fluff. Compare the level of information provided here with Mr. Kallaway's statement about component generation on page A25 of Appendix A.

Mr. Kallaway testified that the "real discussion" in the October 7th Briefing began with his demo, which could not have lasted more than 34 minutes.[225] How could Mr. Kallaway's demo have had significantly more or better information in it than the above-mentioned training video? If this training video contains no trade secrets,[226] how would it have been possible to shoehorn secrets into Mr. Kallaway's demo?

This Alleged Secret Was Already Practiced By HPE

TurnKey has suggested that creating components through a scanning rather than a recording process is part of their alleged secret. What is most puzzling about this allegedly secret process is that cFactory does *not* practice it. cFactory does no recording or scanning. cFactory simply uses an API to invoke UFT in order to do what it is designed to do. How could that possibly constitute a secret? TurnKey adds no value to the generic functionality of UFT. If there is some sort of secret

[224] *Creating Web Components*, TurnKey Solutions (last accessed Feb. 2017), *available at* http://turnkeysolutions.com/wp-content/uploads/2016/03/Creating-Web-Components-Video.wmv
[225] *See* Appendix A, at A4.
[226] Mr. Gannon testified that trade secrets are never told to customers. *See* Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 12

- *HP Business Process Testing for SAP Applications User Guide, v9.5* (Jan. 2008) at 49-50
- *HP Business Process Testing Enterprise Extension User Guide, v10* (Aug. 2009, at 487-488
- *HP Business Process Testing Enterprise Extension User Guide, v11* (March 2011), at 445-446
- *HP Business Process Testing User Guide, v11.5* (June 2012), at 341-343

…which essentially identical to:

- *HP Business Process Testing User Guide, v12.21* (June 2015), at 279-281
- *HP Business Process Testing User Guide, v12.5* (Sept. 2015), at 251-252
- *HP Business Process Testing User Guide, v12.5* (Dec. 2015), at 277-279
- *HP Business Process Testing User Guide, v12.53* (May 2016), at 325-328
- *HP Unified Functional Testing User Guide, v12.54* (Sept. 2016), at 844-856

This process is also disclosed in U.S. Patent Application No. 2010/0325492, filed in 2008. The subject matter of this patent was originally embodied in the BPT For SAP product:

> An embodiment automatically decomposes the application to be tested into discrete, **auto-parameterized test components**, and generates an ordered sequence of components to form a set of logical flows of activity within the application that are subsequently used to test the application.[234]

This Alleged Secret Is Implemented Differently By HPE

HPE's "Smart Record" mode is a renamed version of the "Learn Flow" feature from BPTEE. This feature works—not by using a rules base, as TurnKey's technology does—but rather by receiving information about the application directly from SAP GUI via metadata that SAP provides. This is why the "Smart Record" and "Learn Flow" functionality has been limited to SAP GUI applications.

Analysis Of Secret #5: Proprietary Keyword Methodology/Framework

This Alleged Secret Was Not Specified With Reasonable Particularity

I find this alleged trade secret very puzzling, as TurnKey has been very inconsistent about the scope of its alleged "keyword methodology/framework." Indeed, it appears that TurnKey as dropped this "secret" from the case altogether, as neither of its opening expert reports include it in their respective lists of the asserted trade secrets.[235] I have prepared this section of my report just in case TurnKey changes course again and intends to assert this alleged trade secret at trial.

Critically, TurnKey has never been clear about what it means by "methodology." For example:

> [redacted]
>
> [redacted]

[234] U.S. Patent Application No. 2010/0325492.

[235] Ellis Disclosure, at 14-23; Pedigo Opening Damages Report, at 4-5.

> [redacted].[236]

Below I have analyzed each element of Mr. Ellis' answer:

- [redacted]: Keywords of are made up of code. Mr. Ellis appears to be saying that TurnKey's methodology includes the code that constitutes the product. But TurnKey has not provided code to HPE, nor has it described code in the course of this litigation or alleged that any code was stolen.

- [redacted]: How keywords are developed is a development process; it is know-how. Mr. Ellis appears to be saying that TurnKey's methodology is the methods by which the product is created.

- [redacted]: How the product arranges the keywords when creating components is a behavior of the product. Mr. Ellis seems to be saying that TurnKey's methodology includes the functionality of the product. If so, then "methodology" does not seem to add anything to the other four alleged secrets.

But that is not all. In TurnKey's Supplemental Response to Interrogatory No. 1, it writes: [redacted][237] When asked what this meant, Mr. Ellis replied:

> [redacted].[238]

So, that is a fourth meaning:

- [redacted]: Mr. Ellis seems to be saying that TurnKey's methodology encompasses *anything* that *anyone* does with its product.

In Mr. Ellis' report, he does not specifically define what he means by methodology. But regardless of what he means, his usage of that word is not consistent. Consider these examples:

1. "In Mr. Ellis's experience, competitors often do not make the **design methodologies that underlie their software** public."

2. "This **methodological process and the technology underlying it** was explained in depth during the two HP briefing sessions from October 2014."[239]

In the first case, the methodology is behind the software; in the second case, the software is behind the methodology. Methodology is everywhere. It seems to me that TurnKey uses "methodology" in the same sense as "et cetera"—a catchall to refer to something that, for whatever reason, is left undescribed. This does not satisfy TurnKey's burden to identify its trade secrets with reasonably particularity.

[236] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 136.
[237] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).
[238] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 133.
[239] Ellis Disclosure, at 23.

TurnKey's Response To Interrogatory No. 1 Is Vacuous

In TurnKey's Response to Interrogatory No. 1, the fifth trade secret is described as follows:

> [redacted][240]

I have analyzed this text as carefully as I know how. Although the above excerpt is a lot of text, it specifies very little. This becomes apparent when we unpack the text.

First, I broke out the clauses into an outline form, adding implied words:

[240] TurnKey's Supplemental Response to Interrogatory No. 1 (Feb. 22, 2016).

[redacted]

Figure 49: Hierarchical breakdown of clauses in TurnKey's description of its keyword methodology

The outline clarifies the various clauses and sub-clauses in TurnKey's interrogatory response. Next, I normalized each clause to make it a self-standing assertion of fact. I found 31 assertions total:

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

I see that there are five kinds of assertions in this description:

- **X has the attribute Y** (e.g. "this methodology is keyword-based")
- **X uses Y** (e.g. "this methodology uses auto-parameterization")
- **X includes Y** (e.g. "this methodology includes how parameters are selected.")
- **X might include Y** (e.g. "a test step in this methodology may comprise an output field.")
- **X has the ability to do Y** (e.g. "[the editor] may be shown overlapping an application screen.")

What's missing? *Anything that explains the method inside this "methodology."* A methodology is supposed to be a set of methods for doing something. Yet, I see not a single "how to" in this description. In my professional experience, I would expect to see something like "do this, then that, then look up this value in that table and perform this task with the value…" TurnKey has offered nothing like that. Instead, TurnKey's description refers to phrases like "use cases" without actually providing a use case. It mentions "features" but doesn't say what the features do. Rather than specifying a methodology, this description specifies the length and the breadth—but not the substance—of a territory that TurnKey claims as its intellectual property.

This is not a specification for a methodology. The assertions, here, are placeholders—like the table of contents for a specification that has not yet been written.

The Descriptions of TurnKey's Keyword Methodology By Its Own Employees Are Vacuous

Setting aside TurnKey's Supplemental Response to Interrogatory No. 1, there seem to be no other written materials regarding the TurnKey's keyword methodology. No relevant documents are mentioned in TurnKey's Response to Interrogatory No. 2,[241] and TurnKey has produced no document I am aware of that seems to embody a keyword "methodology." Further, none of TurnKey's witnesses make any reference to a documented methodology during their depositions. The only documents potentially relevant are the various cFactory user guides that describe the keywords it uses.

TurnKey's deposition testimony provides some insight into what TurnKey means by a "methodology," but not much.

References to some sort of methodology abound in Dale Ellis' Rule 30(b)(6) deposition testimony. This is Mr. Ellis' first mention of it:

> [redacted]
>
> [redacted]
>
> [redacted][242]

This sounds fairly general and non-technical. In fact, it is hard to distinguish from TurnKey's own marketing material:

> The first step in automating the testing of an application using cFactory™ is to **capture or learn the screens of the application** using a cFactory™ version of HP's QuickTest Professional (QTP) toolset. **The Analyst launches QTP's 'Add Object' Wizard and clicks on each of the application screens to be automated. This captures or 'learns' the screens, after which the Analyst moves to cFactory™ and clicks the 'Analyze' button at**

[241] HPE's Interrogatory No. 2 asked TurnKey to "identify all documents describing and/or constituting the Trade Secrets."

[242] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 28.

> **the bottom of the window. cFactory™ then analyzes all of the screens, including all of the objects such as fields and buttons that each one contains, etc.** In just seconds, cFactory™ will display all of the automated test components that it can create for the selected screens.
>
> The TDMS™ generator then **creates a complete MS Excel workbook specifically designed and color coded to hold data for that integration test**. Best of all, these data sheets not only store and organize the testing data, but will also data-drive the tests at run time.[243]

TurnKey repeatedly cites its methodology as if it were proprietary, but in what way is it secret or specific?

> [redacted].[244]

Is TurnKey saying here that it has a defined standard, principle or method for designing, grouping and utilizing keywords? If so, it has not described it or shown it to HPE. For example, Mr. Ellis testified:

> [redacted]
>
> [redacted][245]

Mr. Ellis declined to go through all of the examples because of the time it would take, yet TurnKey claims it only took Mr. Kallaway little more than half an hour to convey all five alleged secrets, *of which its allegedly proprietary "keyword methodology" is only one*. Mr. Ellis continued:

> [redacted]
>
> [redacted][246]

Mr. Ellis claims that TurnKey has come up with a "methodological standard and process," but he doesn't say what that process is. He states its purpose but not its essence. He has provided us a label affixed to an empty folder.

Mr. Kallaway speaks of TurnKey's "methodology" in a way that seems to contradict Mr. Ellis:

[243] *TurnKey cFactory Solution Brief*, Orasi (2009), at 2, *previously available at* www.orasi.com/documents/TurnKey-cFactory-Solution-Brief.pdf.
[244] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 135.
[245] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 129.
[246] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 129-130.

> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted][247]

Is TurnKey saying here that "methodology" just means how the software works, and not the standards and principles and methods of how it's designed?

This Alleged Secret Was Already Known To The Industry

To recap, TurnKey seems to say that its proprietary keyword methodology/framework consists of any of these things:

1. The *plainly visible behavior* of Auto-Generation of Components, Auto-Parameterization, Auto-Generation of Data Repositories, and Auto-Detection of Changes as they operate together in TurnKey's product.

2. The *ways users interact* with a specific product that performs Auto-Generation of Components, Auto-Parameterization, Auto-Generation of Data Repositories, and Auto-Detection of Changes.

3. The *design methods* by which TurnKey personnel develop its keyword libraries and rules bases.

[247] Kallaway Deposition (May 17, 2016), at 14.

4. The *software code* embodying specific algorithms that constitute the TurnKey product, including keyword libraries, rules bases, and all other code in the product.

The level of specificity TurnKey uses to describe its methodology is so general and vague that determining whether any element of it is public knowledge is not difficult. If all that's required is a methodology that possesses the attributes listed in TurnKey's Response to Interrogatory No. 1, then anyone having ordinary skill in the art would know how to do that.

Here is a list of core elements of the 31 assertions listed above:

[redacted]

These elements are not described in any substantial detail. In my expert opinion, regardless of whether these elements are stated alone or considered together, they are all publicly known or readily ascertainable.

For instance, "scoping of keywords" is something that a person with ordinary skill in the art would know how to do, because that is an ordinary programming practice. Anyone who designs a system of keywords to control anything must necessarily make design decisions about how to group them and how much scope to give them.

Ryan Jacques discusses the scoping of keywords as follows:

> [redacted][248]

Mr. Jacques appears to be saying that the secret methodology by which keywords are scoped is to "understand the function and then write the code."

This is not a secret. This is ordinary practice. This is what all programmers do.

There is one more source of information about scoping practices: the cFactory User Guide, which is declared non-confidential under the TurnKey OEM agreement and is publicly available on TurnKey's website. Here's what it has to say:

[248] Jacques Deposition (June 8, 2016), at 119.

[redacted]

Figure 50: Scoping practices from the cFactory User Guide[249]

It does appear, from this, that TurnKey's scoping practices are not kept secret. It also appears that these scoping practices are rudimentary suggestions, and would not be considered a technical innovation by anyone skilled in the art.

Setting aside scoping, as I look over the list, on page 75. I do not see *anything* in it which an ordinary programmer wouldn't be able to find a way to implement in the normal course of his work. Of course, again, TurnKey's specific ways of implementing those elements may be special, but if so, they have not disclosed those details.

When considered together, instead of separately, the picture doesn't change. HPE ALM/BPT/UFT is a platform that facilitates doing exactly the kinds of things listed above. Any programmer who has expertise in the HPE platform could create a product with these features on the HPE platform, given reasonable time and motivation to do so. The APIs for that are all documented and provided by HPE for such a purpose.[250]

[249] *TurnKey cFactory 2.6 User Guide: Revision 1,* at 169.

[250] See Appendix C.

The question here is whether HPE personnel, if not given the list of elements above that supposedly constitute TurnKey's methodology, would readily ascertain them. The answer is a resounding yes. The HPE people who create this technology live and breathe automation, components, auto-parameterization, etc. The methodology outlined by TurnKey is very similar to what HPE technology does, not because HPE misappropriated anything, but because HPE invented the basis for the system that TurnKey has built. Indeed, TurnKey's system is an add-on to BPT.

This Alleged Secret Derives No Value From Secrecy

Inasmuch as the methodology includes the visible functionality of the software, and the ways in which the users interact with the software, and the benefits they experience by doing so, TurnKey could not sell its product if it kept those things secret.

The 31 assertions embodied in the description of this alleged secret in TurnKey's Response to Interrogatory No. 1 are too vague to have value as a secret, since they provide no operational guidance on how to manifest any of the capabilities that are mentioned.

This Alleged Secret Was Not Kept Secret By TurnKey

With regard to plainly visible behavior of cFactory and the ways that users interact with cFactory, and given that TurnKey does not restrict the people who are authorized to see its product in action, TurnKey has not kept that aspect of its methodology a secret.

With regard to its code and design methods, TurnKey has never disclosed that information to HPE, either at the October Briefings or during the course of this litigation—it is therefore not an issue.

With regard to keyword code, that information is open to TurnKey's users. In fact, anyone with access to the HPE ALM system into which cFactory or its accelerators is integrated can review it.

TurnKey's YouTube videos, cFactory user manual, and training videos explicitly show the product in action and explain its visible functionality. They are all publicly available. TurnKey provides five days of training to its customers, which cover many aspects of this alleged secret, including what TurnKey now claims is its "methodology" for creating and scoping keywords.[251]

TurnKey's manual describes and shows the format of rules bases and how to edit them.

Simply put, TurnKey has made no effort to maintain the secrecy of the information it has disclosed to HPE during litigation about its alleged keyword methodology.

This Alleged Secret Was Already Practiced By HPE

Anyone who reviews the Business Process Testing user guides, as I have done, will see that HPE specifies a methodology for doing business process testing. This methodology includes keyword-driven testing, creating components, parameterizing them, creating data repositories, and performing change analysis. What HPE specifies are for the most part the same topics that TurnKey specifies as its methodology. The Business Process Testing guides go much further than simply saying that parameterization is done—they explain exactly how to do it. This information is implicitly a methodology, but the manuals also explicitly explain Business Process Testing methodology:

[251] See Appendix B. at B16-17.

Rebuttal Report of James Bach on behalf of Hewlett Packard Enterprise Company

Appendix B:
TurnKey's Public Disclosure Of Its Allegedly Confidential Material

Table of Contents

Turnkey Claims That Its Alleged Trade Secrets Are Known To Only A Few People, Even Within Turnkey . . . . Not So.

Mr. Gannon emphatically testified that TurnKey's confidential information is not widely disseminated, even within TurnKey:

> [redacted].[1]

> [redacted]
>
> [redacted][2]

> [redacted]
>
> [redacted][3]

> [redacted]
>
> [redacted].[4]

> [redacted]
>
> [redacted].[5]

> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted]

[1] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 5.
[2] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 7.
[3] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 8.
[4] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 8.
[5] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 9.

> [redacted][6]

> [redacted]
>
> [redacted][7]

These statements present a picture of *closely held* information—only a few people possess these secrets, and they are understood only by "technical people," by which Mr. Gannon seems to have meant software engineers. The secrets are carefully protected, according to him.

Furthermore, Mr. Gannon's early testimony implies that TurnKey's secrets at issue in this litigation have an *esoteric* nature—i.e. understandable only to engineers. He does not regard TurnKey's secrets as information that anyone can understand, such as a customer list, a server password, or a marketing-level description of the features of a software product. Yet, when TurnKey disclosed its secrets to HPE for the purposes of this litigation, it described high-level, plainly visible features and behaviors, not esoteric technical details.

Mr. Ellis testified that the people who have these secrets are careful about sharing them:

> [redacted][8]

Mr. Gannon later testified, however, that TurnKey's trade secrets might be documented in the user manual—*he doesn't know*:

> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted].[9]

Are the secrets closely held, esoteric technical details? Or are they information about how the product functions, which is freely shared with each customer?

Indeed, I have reviewed the cFactory user guides, and I can confirm that *nearly everything TurnKey describes with respect to its alleged trade secrets is documented in its various user manuals.* In my

[6] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 9.
[7] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 9.
[8] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 147.
[9] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 23.

opinion, there is even more detail about the capabilities of cFactory in the user guides than TurnKey disclosed in its Supplemental Response to Interrogatory No. 1 and in TurnKey's Rule 30(b)(6) deposition on that subject.

Are TurnKey's user manuals confidential? Apparently not.

The user manuals are marked confidential, and Mr. Gannon testified that although users are not subject to a separate confidentiality agreement, the TurnKey user license includes a blanket non-disclosure term.[10] But when I examined the user license during my source code review, I saw that it does not specifically describe the confidential or proprietary material that the user is expected to protect, nor is there evidence that users are provided with any training on this matter. There is also no evidence that TurnKey does any policing or follow-up about its purported confidentiality restrictions.

Furthermore, the user guides are a form of "Documentation" associated with cFactory that is explicitly deemed non-confidential under the OEM Agreement.[11]

Finally, *TurnKey's user manuals are posted publicly on its website.* Anyone can find them with a Google search, as demonstrated below.

Turnkey Claims That Its Personnel Undergo Training To Protect Its Trade Secrets . . . . Not So.

TurnKey's statements about how employees are trained with respect to TurnKey's confidential information are also inconsistent. For instance, Mr. Ellis testified that technical employees are specifically trained in what constitutes company secrets:

> [redacted]
>
> [redacted].[12]

At first Mr. Kallaway seemed to corroborate Mr. Ellis' testimony. But, when pressed for details, he demurred:

> [redacted]

[10] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 23-24.

[11] OEM Agreement §§ 1.3, 10.1 ("Each Program in object code form and related Documentation provided to HP hereunder are deemed non-confidential," where "Documentation" is defined as "manuals and other documentation … [TurnKey] has created for a Program…").

[12] Ellis 30(b)(6) Deposition (Feb. 24, 2016), at 142.

[redacted]

[redacted]

[redacted].[13]

The testimonies of Ms. Karch[14] and Mr. Jacques[15] also indicate that the training given to employees is neither systematic nor specific with regard to what TurnKey's trade secrets actually are. This indicates that employee training covers that topic only in a general, high-level way. This could be sufficient if all genuinely confidential material is clearly marked or otherwise understood to be confidential, and not publicly disclosed. This does not seem to be the case with Ms. Karch, who seemed confused about what it means for something to be confidential:

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[13] Kallaway Deposition (May 17, 2016), at 15.

[14] Karch Deposition (June 30, 2016), at 46. In fact, Ms. Karch could not recall any of TurnKey's alleged trade secrets during her testimony.

[15] Jacques Deposition (June 8, 2016), at 88.

> [redacted]
>
> [redacted]
>
> [redacted]
>
> [redacted][16]

Ms. Karch said she believes there is "confidentiality to TurnKey" in a public presentation given to potential customers who are not under NDA, as long as the slides are marked confidential. Thus, in contrast to claims made during various TurnKey Rule 30(b)(6) depositions, Ms. Karch believes she is not violating any TurnKey policy by sharing TurnKey's confidential information to any random member of the public.

With this, it would seem that TurnKey's training with respect to trade secret protection has not been effective. Yet, Mr. Gannon testified that no TurnKey personnel has been disciplined for improper handling of confidential material.[17]

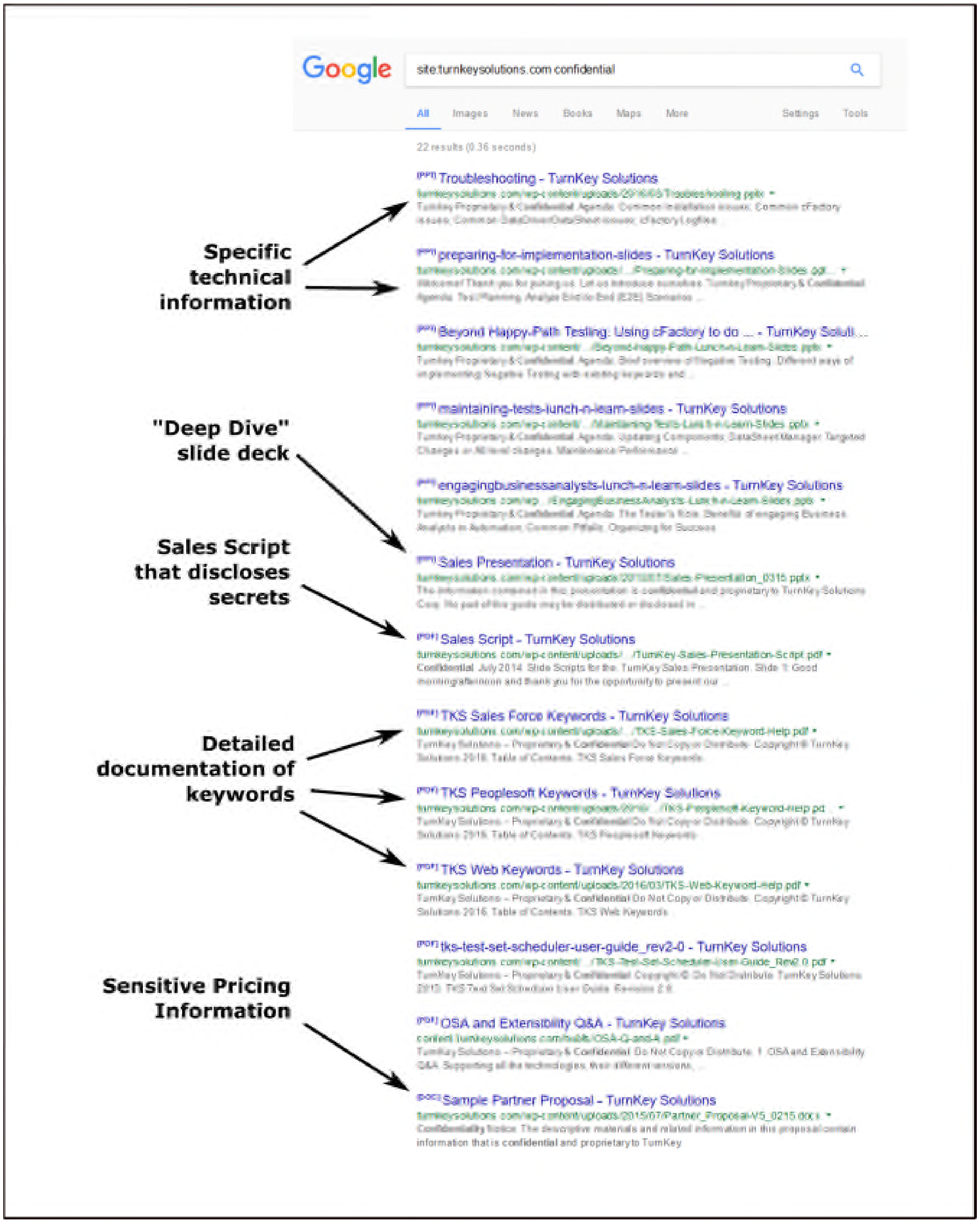
TurnKey Routinely Publishes Confidential Information On Its Own Website

When I started working on this litigation about a year ago, one of the first things I did was to Google TurnKey in order to gather whatever is publicly available about its technology and business. I immediately found documents posted publicly that were identical to documents produced to HPE and designated as "highly confidential."[18]

[16] Karch Deposition (June 30, 2016), at 47.

[17] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 15.

[18] At the time of this writing, the search "site:turnkeysolutions.com confidential" returns 22 hits.


Google
All
Images
News
Books
Maps
More
Settings
Tools
Specific technical information
"Deep Dive" slide deck
Sales Script that discloses secrets
Detailed documentation of keywords
Sensitive Pricing Information

site:turnkeysolutions.com confidential

22 results (0.36 seconds)

[PPT] Troubleshooting - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/2016/05/Troubleshooting.pptx
TurnKey Proprietary & Confidential Agenda: Common installation issues; Common cFactory issues; Common DataDriven/DataSheet issues; cFactory Logfiles ...

[PPT] preparing-for-implementation-slides - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/.../Preparing-for-Implementation-Slides.ppt...
Welcome! Thank you for joining us. Let us introduce ourselves. TurnKey Proprietary & Confidential Agenda: Test Planning; Analyze End to End (E2E) Scenarios ...

[PPT] Beyond Happy-Path Testing: Using cFactory to do ... - TurnKey Soluti...
turnkeysolutions.com/wp-content/.../Beyond-Happy-Path-Lunch-n-Learn-Slides.pptx
TurnKey Proprietary & Confidential Agenda: Brief overview of Negative Testing; Different ways of implementing Negative Testing with existing keywords and ...

[PPT] maintaining-tests-lunch-n-learn-slides - TurnKey Solutions
turnkeysolutions.com/wp-content/.../Maintaining-Tests-Lunch-n-Learn-Slides.pptx
TurnKey Proprietary & Confidential Agenda: Updating Components; DataSheetManager; Targeted Changes vs All-level changes; Maintenance Performance ...

[PPT] engagingbusinessanalysts-lunch-n-learn-slides - TurnKey Solutions
turnkeysolutions.com/wp.../EngagingBusinessAnalysts-Lunch-n-Learn-Slides.pptx
TurnKey Proprietary & Confidential Agenda: The Tester's Role; Benefits of engaging Business Analysts in Automation; Common Pitfalls; Organizing for Success

[PPT] Sales Presentation - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/2015/07/Sales-Presentation_0315.pptx
The information contained in this presentation is confidential and proprietary to TurnKey Solutions Corp. No part of this guide may be distributed or disclosed in ...

[PDF] Sales Script - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/.../TurnKey-Sales-Presentation-Script.pdf
Confidential. July 2014. Slide Scripts for the. TurnKey Sales Presentation. Slide 1: Good morning/afternoon and thank you for the opportunity to present our ...

[PDF] TKS Sales Force Keywords - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/.../TKS-Sales-Force-Keyword-Help.pdf
TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute. Copyright © TurnKey Solutions 2016. Table of Contents. TKS Sales Force Keywords.

[PDF] TKS Peoplesoft Keywords - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/2016/.../TKS-Peoplesoft-Keyword-Help.pd...
TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute. Copyright © TurnKey Solutions 2016. Table of Contents. TKS Peoplesoft Keywords.

[PDF] TKS Web Keywords - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/2016/03/TKS-Web-Keyword-Help.pdf
TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute. Copyright © TurnKey Solutions 2016. Table of Contents. TKS Web Keywords.

[PDF] tks-test-set-scheduler-user-guide_rev2-0 - TurnKey Solutions
turnkeysolutions.com/wp-content/.../TKS-Test-Set-Scheduler-User-Guide_Rev2.0.pdf
TurnKey Solutions – Proprietary & Confidential Copyright © Do Not Distribute. TurnKey Solutions 2013. TKS Test Set Scheduler User Guide. Revision 2.0.

[PDF] OSA and Extensibility Q&A - TurnKey Solutions
content.turnkeysolutions.com/hubfs/OSA-Q-and-A.pdf
TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute. 1. OSA and Extensibility Q&A. Supporting all the technologies, their different versions, ...

[DOC] Sample Partner Proposal - TurnKey Solutions
turnkeysolutions.com/wp-content/uploads/2015/07/Partner_Proposal-V5_0215.docx
Confidentiality Notice: The descriptive materials and related information in this proposal contain information that is confidential and proprietary to TurnKey

Figure 1: Results of a Google search for "site:turnkeysolutions.com confidential"

Using the search term "site:turnkeysolutions.com confidential," I found numerous documents that I would have expected TurnKey to consider confidential, and in fact are marked confidential. One of the files I found on the site was almost identical to the slide deck used during the October Briefings.[19] This slide deck, in particular, has a full slide dedicated to confidentiality. However, that slide is marked "hidden" in the version of the slide deck now publicly posted on TurnKey's website. This implies that *sometimes* TurnKey shows the slide and *sometimes* it doesn't. This also implies TurnKey's employees

[19] The public version of this slide deck was only missing the WorkSoft competitive analysis slides.

treat the *same* information as either confidential or non-confidential depending on the situation. That is not how confidentiality works.

The amount of seemingly confidential TurnKey material available online has increased over the course of this year. By June, I had found 313 pages of confidential material. By October, that had grown to 1,462 pages.

Below is the complete list of the files I discovered publicly that are designated as confidential by TurnKey. All but one of these documents are posted on TurnKey's own website. Therefore, this is not a case of unauthorized third-party publication.

Note that I performed no "hacking" to see these files. I did not log into any website with false or stolen credentials. I did not violate any posted Terms Of Use policy, either (Indeed TurnKey's website has no Terms Of Use posted.) I simply Googled and browsed all the public-facing material on TurnKey's site:

"Confidential" TurnKey Documents That Are Publicly Available

File Name	Confidentiality Designation
ALM-Best-Practices091515.pdf	"TURNKEY PROPRIETARY & CONFIDENTIAL"
Automated-Testing-w-Regulatory-Requirements.pptx	"TURNKEY PROPRIETARY & CONFIDENTIAL"
Beyond-Happy-Path-Lunch-n-Learn-Slides.pptx	"Turnkey Proprietary & Confidential"
EngagingBusinessAnalysts-Lunch-n-Learn-Slides.pptx	"Turnkey Proprietary & Confidential"
Maintaining-Tests-Lunch-n-Learn-Slides.pptx	"Turnkey Proprietary & Confidential"
OSA-QandA.docx	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
Partner_Proposal-V5_0215.docx	"The descriptive materials and related information in this proposal contain information that is confidential and proprietary to TurnKey. No rights in said material are hereby transferred to Customer. This information is submitted with the express understanding that it will be held in strict confidence and will not be disclosed, duplicated or used, in whole or in part, for any purpose other than the evaluation of this proposal."
Preparing for TurnKey Implementation and Training_v2.2.pdf	Confidentiality Notice: The descriptive materials and related information in this document contain information that is confidential and proprietary to TurnKey. No rights in said material are hereby transferred to TurnKey's Partners or Customers. This information is submitted with the express understanding that it will be held in strict confidence and will not be disclosed, duplicated or used, in whole or in part, for any purpose other than for information only.
Preparing-for-Implementation-Slides.pptx	"Turnkey Proprietary & Confidential"
Putty Bulk Component Creation-Final.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Turnkey Solutions."
Putty Log File Flush-Final.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Turnkey Solutions."
Putty Single Component Creation-Final.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Turnkey Solutions."
Sales-Presentation_0315.pptx	"The information contained in this presentation is confidential and proprietary to TurnKey Solutions Corp. No part of this guide may be distributed or disclosed in any form to any third party without written permission by TurnKey."
TE.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without

	the prior written consent of Turnkey Solutions."
TE_TKS.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Turnkey Solutions."
TKS Object Identification QTP.vbs	"This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Turnkey Solutions."
TKS-Web-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-Oracle-JInit-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-Oracle-OAF-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-Peoplesoft-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-Sales-Force-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-SAP-GUI-Keyword-Help.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TKS-Test-Set-Scheduler-User-Guide_Rev2.0.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
Troubleshooting.pptx	"Turnkey Proprietary & Confidential"
TurnKey-Training-Agenda-2016.pdf	"TurnKey Solutions – Proprietary & Confidential Do Not Copy or Distribute"
TurnKey_cF_2_6_User_Guide_Rev2.doc	Confidentiality Notice: The descriptive materials and related information in this document contain information that is confidential and proprietary to TurnKey. No rights in said material are hereby transferred to TurnKey's Partners or Customers. This information is submitted with the express understanding that it will be held in strict confidence and will not be disclosed, duplicated or used, in whole or in part, for any purpose other than for information only.
TurnKey_LunchnLearn_TestMetrics_033116.pptx	"The information contained in this presentation is confidential and proprietary to TurnKey Solutions Corp. No part of this document may be distributed or disclosed in any form to any third party without written permission by TurnKey Solutions. The information herein is subject to change without notice. The only warranties for TurnKey Solutions products and services are set forth in the express warranty statements accompanying such products and services. ©Copyright 2016 TurnKey Solutions Corp. cFactory™, dataDriver™ and the Accelerator logos are trademarks of TurnKey Solutions. All rights reserved."
TurnKey-Business-Case-Analysis_Master-05042015BO.xlsx	(There is no designation in the document, but specific TurnKey pricing information is included in the document, which I infer is intended to be TurnKey confidential)
TurnKey-Keyword-Libraries-and-Definitions.pptx	"Turnkey Proprietary & Confidential"
TurnKey-Sales-Presentation-Script.pdf	"TurnKey Solutions Corp. Confidential"
VIVIT_webinar_3_3_15_Final.pdf	"The information contained in this presentation is confidential and proprietary to TurnKey Solutions Corp."

My ability to find these materials so easily contradicts the testimony of TurnKey's deposition witnesses who indicated that information designated as confidential is not released to the public. To the contrary, it is consistent with a lax attitude towards policing its confidentiality policies. Mr. Gannon exemplified this attitude when he testified on behalf of TurnKey that no one at the company regularly engages in so-called "self-Googling," one of the simplest practices for preventing the unauthorized distribution of confidential documents:

> [redacted].[20]

Turnkey Routinely Reveals Its Alleged Secrets In Youtube Videos

I reviewed 14 videos posted on YouTube that describe the cFactory product, plus one more on Vimeo.com. An interesting pattern emerged right away: the older videos have more information than the newer ones. In fact, the oldest video I could find was a cFactory demo from 2013 given by Stephen Rosing. It was very informative. Despite not being marked by TurnKey as confidential, it demonstrates many of the capabilities asserted as trade secrets in this case.

The earliest five videos all have product demos in them, but four out of five of those videos have no confidential markings. An interesting change hits around June of 2014 (near the time that TurnKey filed a provisional patent application), because all the videos after that date include no demonstration of the actual product, and all of them are marked confidential. Many of the videos share slides with the deck used in the October Briefings, as I show in the slide-by-slide analysis. But that changes again after TurnKey sent its demand letter to HPE in June 2015. The videos posted after that time no longer use the same slides as were used in the October Briefings. It seems that deck was retired, at that time, from recorded *public* presentations. All of this suggests an inconsistent strategy about sharing information publicly about cFactory, at least on YouTube. Regardless, the old videos are still out there and available, and TurnKey is still giving sales demonstrations at major conferences.

Many of the videos include Q&A sessions. I have not come across any instance in which a TurnKey representative has refused to answer a question on the grounds of confidentiality. In many cases the TurnKey representative offers in-depth live demonstrations of the product to anyone who wants one.

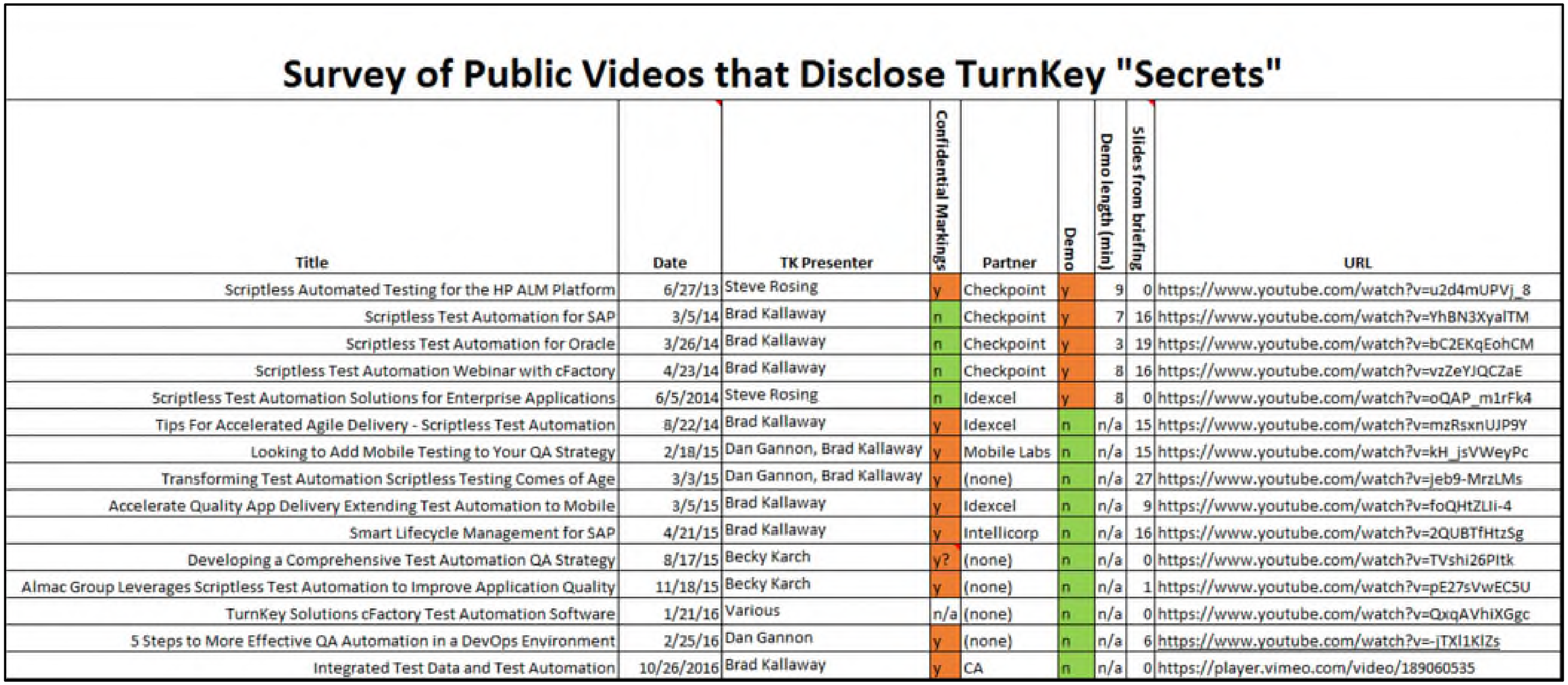
> Q. The next question is: on the demo we did not see anything related to verification. How will they do verification on cFactory? Stephen Rosing: It was just a matter of time. **Typically, I like to do about a 90-minute demo** in order to show all of the capabilities. **In fact, a lot of the questions that were brought up are typically answered in the full demo…”**[21]

My understanding of confidentiality markings from this video is that they are normally used to explicitly classify material as—at the very least—not appropriate for public disclosure. In its videos on YouTube (and one that I located on Vimeo.com), TurnKey uses its confidentiality markings as a sort of optional protection that kicks in whenever TurnKey decides it needs it. This is obviously incompatible with reasonable maintenance of trade secrets.

My research on TurnKey's publicly available demo videos follows:

[20] Gannon 30(b)(6) Deposition (Feb. 24, 2016), at 26.

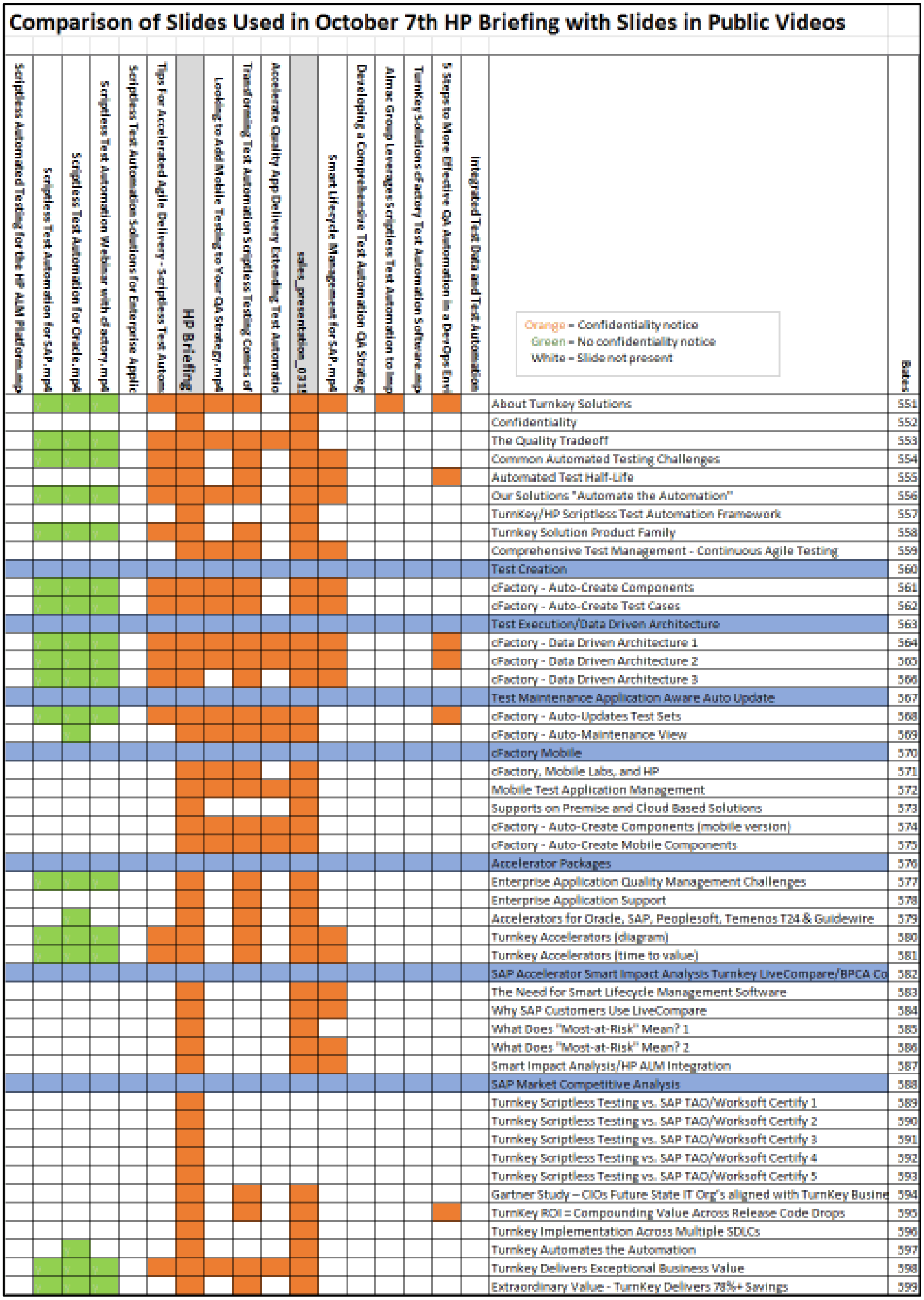
[21] Stephen Rosing, *Scriptless Automated Software Testing for the HP ALM Platform*, YOUTUBE (June 27, 2013), *available at* https://www.youtube.com/watch?v=u2d4mUPVj_8.



Survey of Public Videos that Disclose TurnKey "Secrets"

Title	Date	TK Presenter	Confidential Markings	Partner	Demo	Demo length (min)	Slides from briefing	URL
Scriptless Automated Testing for the HP ALM Platform	6/27/13	Steve Rosing	y	Checkpoint	y	9	0	https://www.youtube.com/watch?v=u2d4mUPVj_8
Scriptless Test Automation for SAP	3/5/14	Brad Kallaway	n	Checkpoint	y	7	16	https://www.youtube.com/watch?v=YhBN3XyalTM
Scriptless Test Automation for Oracle	3/26/14	Brad Kallaway	n	Checkpoint	y	3	19	https://www.youtube.com/watch?v=bC2EKqEohCM
Scriptless Test Automation Webinar with cFactory	4/23/14	Brad Kallaway	n	Checkpoint	y	8	16	https://www.youtube.com/watch?v=vzZeYJQCZaE
Scriptless Test Automation Solutions for Enterprise Applications	6/5/2014	Steve Rosing	n	Idexcel	y	8	0	https://www.youtube.com/watch?v=oQAP_m1rFk4
Tips For Accelerated Agile Delivery - Scriptless Test Automation	8/22/14	Brad Kallaway	y	Idexcel	n	n/a	15	https://www.youtube.com/watch?v=mzRsxnUJP9Y
Looking to Add Mobile Testing to Your QA Strategy	2/18/15	Dan Gannon, Brad Kallaway	y	Mobile Labs	n	n/a	15	https://www.youtube.com/watch?v=kH_jsVWeyPc
Transforming Test Automation Scriptless Testing Comes of Age	3/3/15	Dan Gannon, Brad Kallaway	y	(none)	n	n/a	27	https://www.youtube.com/watch?v=jeb9-MrzLMs
Accelerate Quality App Delivery Extending Test Automation to Mobile	3/5/15	Brad Kallaway	y	Idexcel	n	n/a	9	https://www.youtube.com/watch?v=foQHtZLIi-4
Smart Lifecycle Management for SAP	4/21/15	Brad Kallaway	y	Intellicorp	n	n/a	16	https://www.youtube.com/watch?v=2QUBTfHtzSg
Developing a Comprehensive Test Automation QA Strategy	8/17/15	Becky Karch	y?	(none)	n	n/a	0	https://www.youtube.com/watch?v=TVshi26PItk
Almac Group Leverages Scriptless Test Automation to Improve Application Quality	11/18/15	Becky Karch	y	(none)	n	n/a	1	https://www.youtube.com/watch?v=pE27sVwEC5U
TurnKey Solutions cFactory Test Automation Software	1/21/16	Various	n/a	(none)	n	n/a	0	https://www.youtube.com/watch?v=QxqAVhiXGgc
5 Steps to More Effective QA Automation in a DevOps Environment	2/25/16	Dan Gannon	y	(none)	n	n/a	6	https://www.youtube.com/watch?v=-jTXl1KIZs
Integrated Test Data and Test Automation	10/26/2016	Brad Kallaway	y	CA	n	n/a	0	https://player.vimeo.com/video/189060535

Figure 2: Survey of public videos that disclose TurnKey's "Secrets"



Comparison of Slides Used in October 7th HP Briefing with Slides in Public Videos

Scriptless Automated Testing for the HP ALM Platform.mp	Scriptless Test Automation for SAP.mp4	Scriptless Test Automation for Oracle.mp4	Scriptless Test Automation Webinar with cFactory.mp4	Scriptless Test Automation Solutions for Enterprise Applic	Tips For Accelerated Agile Delivery - Scriptless Test Autom	HP Briefing	Looking to Add Mobile Testing to Your QA Strategy.mp4	Transforming Test Automation Scriptless Testing Comes of	Accelerate Quality App Delivery Extending Test Automatio	sales_presentation_031	Smart Lifecycle Management for SAP.mp4	Developing a Comprehensive Test Automation QA Strateg	Almac Group Leverages Scriptless Test Automation to Imp	TurnKey Solutions cFactory Test Automation Software.mp	5 Steps to More Effective QA Automation in a DevOps Envi	Integrated Test Data and Test Automation	🟧 Orange = Confidentiality notice Green = No confidentiality notice White = Slide not present	Bates
	y	y	y		🟧	🟧	🟧	🟧		🟧	🟧		🟧		🟧		About Turnkey Solutions	551
						🟧				🟧							Confidentiality	552
	y	y	y		🟧	🟧	🟧	🟧	🟧	🟧							The Quality Tradeoff	553
	y	y	y		🟧	🟧		🟧		🟧	🟧						Common Automated Testing Challenges	554
					🟧	🟧		🟧		🟧	🟧				🟧		Automated Test Half-Life	555
	y	y	y		🟧	🟧	🟧	🟧		🟧	🟧						Our Solutions "Automate the Automation"	556
						🟧				🟧							TurnKey/HP Scriptless Test Automation Framework	557
	y	y	y		🟧	🟧		🟧		🟧							Turnkey Solution Product Family	558
						🟧	🟧	🟧		🟧	🟧						Comprehensive Test Management - Continuous Agile Testing	559
																	Test Creation	560
	y	y	y		🟧	🟧	🟧	🟧		🟧	🟧						cFactory - Auto-Create Components	561
	y	y	y		🟧	🟧	🟧	🟧		🟧	🟧						cFactory - Auto-Create Test Cases	562
																	Test Execution/Data Driven Architecture	563
	y	y	y		🟧	🟧	🟧	🟧	🟧	🟧	🟧				🟧		cFactory - Data Driven Architecture 1	564
	y	y	y		🟧	🟧	🟧	🟧	🟧	🟧	🟧				🟧		cFactory - Data Driven Architecture 2	565
	y	y	y		🟧	🟧		🟧		🟧	🟧						cFactory - Data Driven Architecture 3	566
																	Test Maintenance Application Aware Auto Update	567
	y	y	y		🟧	🟧	🟧	🟧	🟧	🟧					🟧		cFactory - Auto-Updates Test Sets	568
		y				🟧	🟧	🟧	🟧	🟧							cFactory - Auto-Maintenance View	569
																	cFactory Mobile	570
						🟧	🟧	🟧		🟧							cFactory, Mobile Labs, and HP	571
						🟧	🟧	🟧	🟧	🟧							Mobile Test Application Management	572
						🟧				🟧							Supports on Premise and Cloud Based Solutions	573
						🟧	🟧	🟧	🟧	🟧							cFactory - Auto-Create Components (mobile version)	574
						🟧	🟧	🟧	🟧	🟧							cFactory - Auto-Create Mobile Components	575
																	Accelerator Packages	576
	y	y	y			🟧		🟧		🟧							Enterprise Application Quality Management Challenges	577
						🟧		🟧		🟧							Enterprise Application Support	578
		y				🟧		🟧		🟧							Accelerators for Oracle, SAP, Peoplesoft, Temenos T24 & Guidewire	579
	y	y	y		🟧	🟧		🟧		🟧	🟧						Turnkey Accelerators (diagram)	580
	y	y	y		🟧	🟧		🟧		🟧	🟧						Turnkey Accelerators (time to value)	581
																	SAP Accelerator Smart Impact Analysis Turnkey LiveCompare/BPCA Co	582
						🟧				🟧	🟧						The Need for Smart Lifecycle Management Software	583
						🟧				🟧	🟧						Why SAP Customers Use LiveCompare	584
						🟧				🟧							What Does "Most-at-Risk" Mean? 1	585
						🟧				🟧	🟧						What Does "Most-at-Risk" Mean? 2	586
						🟧				🟧	🟧						Smart Impact Analysis/HP ALM Integration	587
																	SAP Market Competitive Analysis	588
						🟧											Turnkey Scriptless Testing vs. SAP TAO/Worksoft Certify 1	589
						🟧											Turnkey Scriptless Testing vs. SAP TAO/Worksoft Certify 2	590
						🟧											Turnkey Scriptless Testing vs. SAP TAO/Worksoft Certify 3	591
						🟧											Turnkey Scriptless Testing vs. SAP TAO/Worksoft Certify 4	592
						🟧											Turnkey Scriptless Testing vs. SAP TAO/Worksoft Certify 5	593
						🟧		🟧		🟧							Gartner Study – CIOs Future State IT Org's aligned with TurnKey Busine	594
						🟧		🟧		🟧					🟧		TurnKey ROI = Compounding Value Across Release Code Drops	595
						🟧				🟧							Turnkey Implementation Across Multiple SDLCs	596
		y				🟧				🟧							Turnkey Automates the Automation	597
	y	y	y		🟧	🟧	🟧	🟧	🟧	🟧							Turnkey Delivers Exceptional Business Value	598
	y	y	y			🟧		🟧		🟧							Extraordinary Value - TurnKey Delivers 78%+ Savings	599

Figure 3: Comparison of slides used in October 7th HP briefing with slides in public videos

In 2014, Turnkey Created Unclassified Training Videos For Its Customers That Disclosed Many Of The "Secrets" And Subsequently Posted Them For Public View

As I write this report, five videos are available on TurnKey's website which explain cFactory and the TurnKey Accelerators in detail:

- The videos include more than two hours of detailed demonstration of TurnKey technology (no slides), compared to less than an hour for the combined briefings given to for HPE.
- A transcript of the videos showed more than 18,000 spoken words of explanation.
- There are no confidentiality markings on the videos. Nor are there any copyright notices. Nor does the presenter orally mention anything about confidentiality.
- Statements on the videos indicate that they were prepared for customers, but they are actually available to anyone.
- The videos were recorded by Melissa Jansen, a (now former) TurnKey employee whose job description included "preparation of training videos and documentation, implementation of customer test scripts for training examples, training the customers on test automation using TurnKey's proprietary software..."
- The videos were created *before* the HP briefings of October 2014.

TurnKey Training Videos That Are Publicly Available

Video	Date	Length	Topics
Common-Web-Components-and-Test-Cases-Video-1.wmv	7/29/2014	0:23:10	Covers how to use the Oracle Accelerator for web apps. Mentions how the data sheet creator needs certain information in order to properly name the columns in the data sheet. Explains methodology.
Creating-Web-Components-Video.wmv	7/29/2014	0:26:27	Shows and explains keywords. Shows auto-parameterized steps. Shows component editor in detail. Shows the process of creating components. Mentions rules bases. Explains some things about how the data sheet creator works. Gives details about how to work with tables on web pages. Explains methodology.
Creating-Test-Cases.wmv	7/30/2014	0:13:11	Covers how to use HP ALM/BPT to construct test cases using the TurnKey accelerator components. Discusses how datasheets interact with the test cases.
DataSheets-and-Test-Sets-Video.wmv	7/31/2014	0:40:15	Covers many details about the way datasheets work. Covers parameters and how they work. Mentions that the names of components are used in the datasheet and the names of parameters are column names.
Complex-Flow.wmv	9/19/2014	0:28:12	Covers details about how datasheets are interpreted for testing screens that have tables. Covers details about the algorithm by which cFactory Dataloader will use data from the datasheet in running tests.

The videos constitute exactly the sort of review that HPE asked for and which Brad Kallaway appears to have given, except these videos provide more depth—collectively, they are almost four times longer than his demo. *Auto-Detection of Changes* is not covered, and little is said about how the rules bases work. But what *is* covered in detail includes the alleged trade secrets of *Data Repository Creation*, and Ke*yword Methodology*. *Component Generation* and *Auto-Parameterization* are evident. The TurnKey Component Editor, cited as part of the methodology secret, is in plain view and its use is explained in detail.

The Training Videos Contain Alleged Secrets. Example: Column Names

TurnKey has several times made the claim that one of its "secrets" is the fact that the columns in datasheets have the same name as the parameters they represent. But TurnKey personnel openly speak of column naming in their videos:

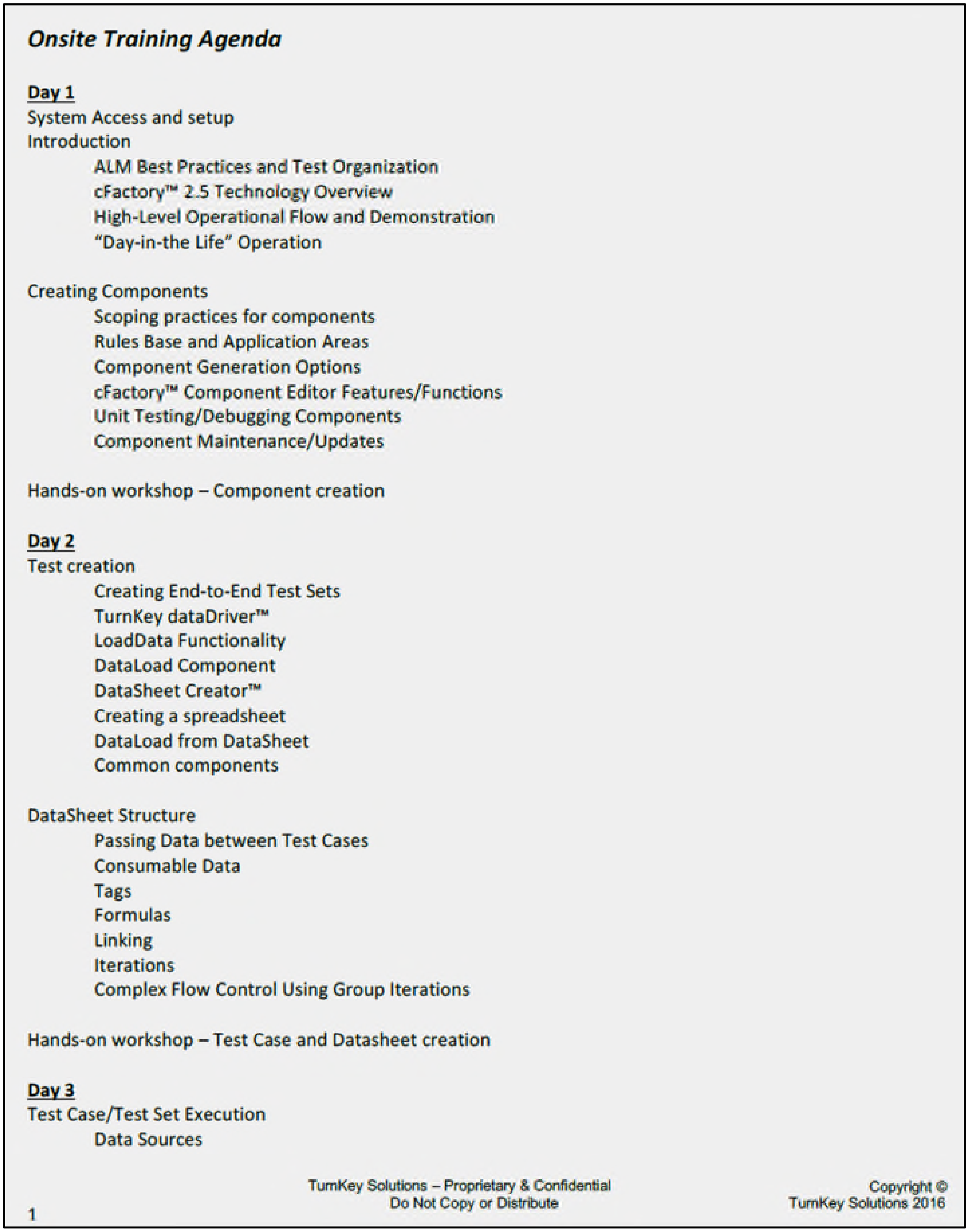
> The next thing I want to point out with the datasheets is the actual structure on the tabs for the test cases. **The first thing to note is that along the top in row number two, you have the list of the component names.** The components that are inside the test case are all listed along the top row in row number two. You'll see their component names are inside the gray borders, and it's a single merged cell for the entire component in row number two. If I scroll back to the beginning, you can see here, and **then if I go to that test case in Quality Center, you'll see the matching components in the test script.** You'll see capture text field web, navigator forms, find mass additions, capture text field web, navigator forms, find mass additions and it'll continue on through the rest of the test case. The next thing to notice is in row number three. **Underneath each component title, you have these cells that represent the parameters or the individual fields on the screen. These are the names of the values on your screen or the fields on your screen.**[22]

TurnKey Gives Five Days Of In-Depth Training To Its Customers, Which Includes Topics It Alleges Are Trade Secrets

I found the agenda below on TurnKey's website. Despite being marked confidential, it is available through a routine Google search. The agenda lays out five days of training. This is truly a "deep dive," compared to the 34-minute product demonstration that HPE received.

The agenda includes material about rules bases, scoping practices, component creation, component editor, creating a spreadsheet, how their data loader interacts with the spreadsheet, datasheet structure, not just using keywords but how to create them. All of these topics are incorporated into TurnKey's alleged trade secrets, as defined TurnKey's Supplemental Response to Interrogatory No. 1. Yet, TurnKey appears to be sharing them freely.

[22] *DataSheets and Test Sets* (July 31, 2014)*, available at* http://turnkeysolutions.com/wp-content/uploads/2016/03/DataSheets-and-Test-Sets-Video.wmv



Onsite Training Agenda

Day 1

System Access and setup

Introduction

- ALM Best Practices and Test Organization
- cFactory™ 2.5 Technology Overview
- High-Level Operational Flow and Demonstration
- "Day-in-the Life" Operation

Creating Components

- Scoping practices for components
- Rules Base and Application Areas
- Component Generation Options
- cFactory™ Component Editor Features/Functions
- Unit Testing/Debugging Components
- Component Maintenance/Updates

Hands-on workshop – Component creation

Day 2

Test creation

- Creating End-to-End Test Sets
- TurnKey dataDriver™
- LoadData Functionality
- DataLoad Component
- DataSheet Creator™
- Creating a spreadsheet
- DataLoad from DataSheet
- Common components

DataSheet Structure

- Passing Data between Test Cases
- Consumable Data
- Tags
- Formulas
- Linking
- Iterations
- Complex Flow Control Using Group Iterations

Hands-on workshop – Test Case and Datasheet creation

Day 3

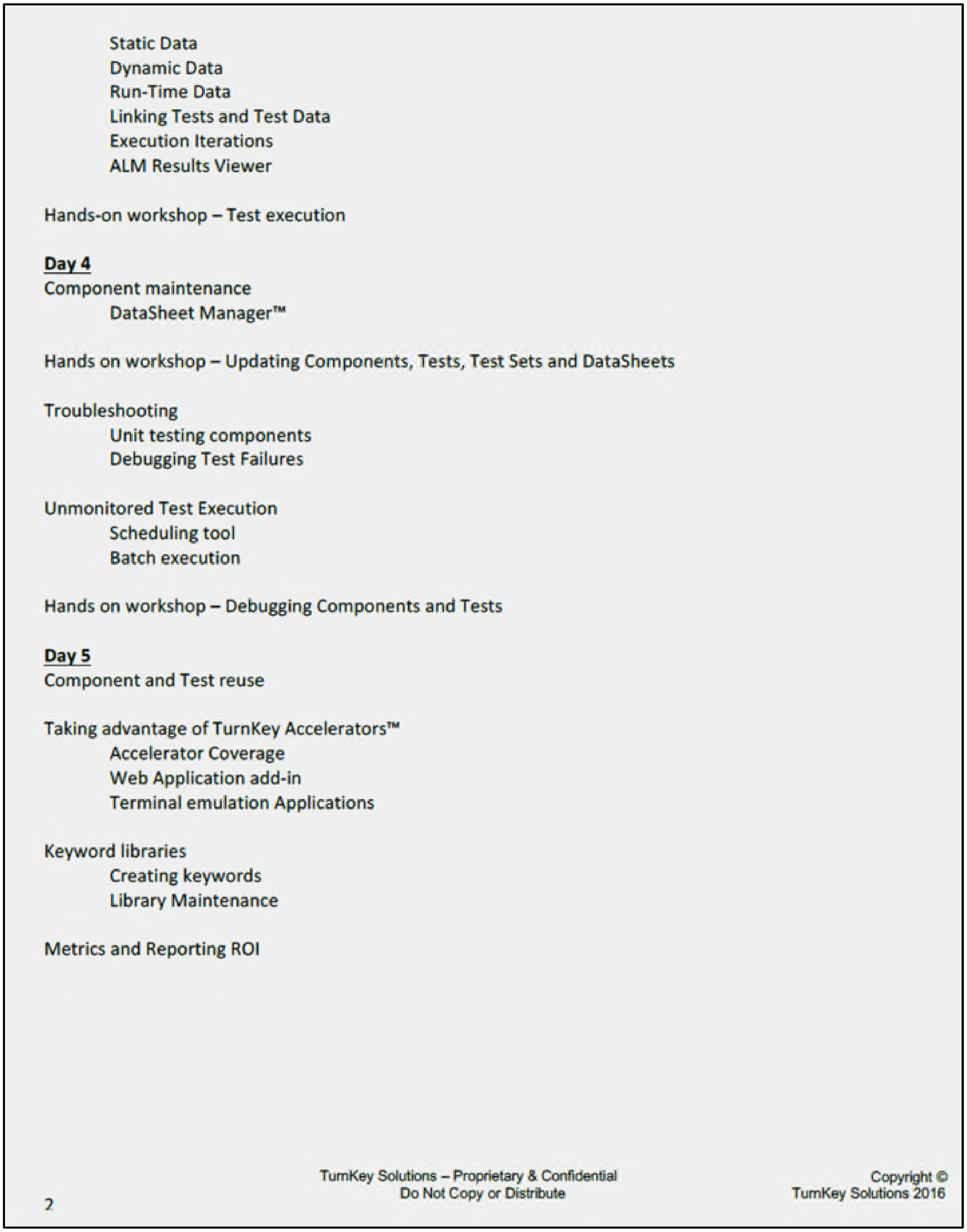
Test Case/Test Set Execution

- Data Sources

TurnKey Solutions – Proprietary & Confidential
Do Not Copy or Distribute

1

Figure 4: TurnKey's "Confidential" Onsite Training Agenda for Customers, Page 1



Static Data
Dynamic Data
Run-Time Data
Linking Tests and Test Data
Execution Iterations
ALM Results Viewer

Hands-on workshop – Test execution

Day 4
Component maintenance
DataSheet Manager™

Hands on workshop – Updating Components, Tests, Test Sets and DataSheets

Troubleshooting
Unit testing components
Debugging Test Failures

Unmonitored Test Execution
Scheduling tool
Batch execution

Hands on workshop – Debugging Components and Tests

Day 5
Component and Test reuse

Taking advantage of TurnKey Accelerators™
Accelerator Coverage
Web Application add-in
Terminal emulation Applications

Keyword libraries
Creating keywords
Library Maintenance

Metrics and Reporting ROI

TurnKey Solutions – Proprietary & Confidential
Do Not Copy or Distribute

Copyright ©
TurnKey Solutions 2016

2

Figure 5: TurnKey's "Confidential" Onsite Training Agenda for Customers, Page 2

Based on the information provided in this Appendix B, it is clear to me that TurnKey has not taken reasonable precautions to maintain the secrecy of its alleged trade secrets.