

## Chapter 8: Another Day Another Dollar

In the morning after breakfast Sandi and Lionel drive to Claude's Café for the morning coffee. During the social coffee hour, members of the brew crew gently teased Lionel about his black smartpad bag.

"Whatcha got in your purse, Lionel?", ask Smoooth . Smoooth knows that Lionel is "carrying" today for some reasons.

"Two points, Smoooth," raising two fingers like a wrestling referee Boilermaker chimes-in.

"I needed something to carry all that money that I receive from my many fishing lure customers," replies Lionel. He then adds "'va bene, va bene'" to let Smoooth know that all is well. Lionel had recent help prepare Smoooth for a trip to Italy. So now Smoooth knows that "It all goes well."

Lionel and Sandi gamely endured the jocular banter. Eventually the topic baton is passed to the next member of the brew crew.

Returning home Sandi and Lionel slipped into their retirement routine. Following his cycle ride and personal care routine, he is heads to the workspace/office.

Just before restarting where he finished last night Lionel scans his calendar observing that in two today's time he has a late morning appointment at the Durham VA hospital for some x-rays of an old military service injury. Noting that appointment, he texts Sandi asking if she would like to go shopping at Brightleaf in Durham. He anticipates that the x-ray appointment should be a quick in/out event.

Finally returning to the Frank saga he starts the audio/video equipment and begins. "We will be looking in depth at the contents of the 'eclipse\_devwrk/trunk/e4Spn' svn project." With that declaration Lionel opens a web session and enters '<http://localhost/svn/>'. He then navigates down the chain to [http://localhost/svn/eclipse\\_devwrk/trunk/e4Spn/](http://localhost/svn/eclipse_devwrk/trunk/e4Spn/).

"The e4Spn project is composed o six sub-folders: bin, build, com, docs, lib:, web. The e4Spn folders bin, docs, and lib are currently empty.

"Scanning the 'svn' web page presentation of the 'Spinnaker' project we will work down the pathway that makes good engineering sense to us. We will focus on the build, com, and web folders. Let's briefly discuss on a high level what these three folders contain before jumping into the the technical details. The 'build folder' contains files that are used to assemble artifacts, create executable application code, and package web documents. The 'com' folder' contains the software code which Frank and his software engineering colleagues created for the Spinnaker application. The 'web folder' is a complete rendering of the Spinnaker web application artifacts as they 'would appear' on a server hosting the application.

Now let's put our software engineering miner's helmet to begin looking at the technical details. Since the 'Spinnaker' project 'builds' a web application we will start with the build folder. The /trunk/e4Sys/build folder contains two critical files: build.properties and build.xml. The build.properties file has a simple list of definitions that link the 'Spinnaker' project to global variables. All of Frank's definitions are within normal expectations and practices.

"The build.xml file is a different 'dog'. The build.xml is a collection of rules expressed as variables definitions called properties and process step definitions. As such the build.xml is sometimes called the build rulescontainer. The ordering of the process steps is critical. The awkwardness of having to employ the web version of the svn is obvious at this point. What we want to do is view the [http://localhost/svn/eclipse\\_devwrk/trunk/e4Sys/build/build.xml](http://localhost/svn/eclipse_devwrk/trunk/e4Sys/build/build.xml) page to investigate the build rules by following a variable or process step through the other artifacts within the svn. To do that we open another session or tab view of [http://localhost/svn/eclipse\\_devwrk/trunk/e4Spn/](http://localhost/svn/eclipse_devwrk/trunk/e4Spn/). Other than mentally confuse, there is no problem with collisions or overwriting the svn controlled artifacts.

"Sequentially inspecting the first twelve properties produced no surprise. But when we reach further we begin to see 'the soft under belly' of the Spinnaker project. The first twelve properties of the build rules are used to establish new handles to be used within the build process. The next seven properties setup linkages between 'Spinnaker' web application artifacts and the name to which they be referenced during the build activity. Properties are stated using a technique called name and value pairs. For example one property name is src.vm.dir'. Its assigned value is '\${web.dir}/templates'.

"From within the 'eclipse' environment Frank would invoke the 'java compiler' from an automated build tool called 'ant'. The 'ant' tool creeps through the build.properties and build.xml files and creates at least one java applet file. It subsequently, pushes various run time artifacts to the locations defined in the build.xml file to their appropriate locations. The beauty of this approach is that it repeatable. Once the contents of the build.properties and build.xml are defined and tested the results will only change in the manner a software engineer has specified. The ugly part is the complexity and having to troubleshoot "almost human readable" statements.

"We have extensive experience with this environment, but diving deeper probably is not warranted at this time.

"We are now going to concentrate on the 'com folder'. The 'com folder' provides us a complete listing of all the files down to the lowest level. Fortunately, for us, Frank has all the file artifacts in the svn repository that he created and update. Plus he has pulled the work of his fellow engineers so that he can 'test' his work. We can see all those project files from the [http://localhost/svn/eclipse\\_devwrk/trunk/e4Spn/](http://localhost/svn/eclipse_devwrk/trunk/e4Spn/) web page. That will require us to hunt and peck through the code collection to see what we can learn. If we 'discover a real need' we can we 'check out' the entire body of artifacts from the svn or subset. Then we can load them into a work folder to examine them further detail. This would be tedious.

"Finally we will laser in on the 'web folder'. In this folder we find details of coding conventions, project architecture, and project structure. We discover collection of 117 web pages used to gather inputs or to provide outputs to the Spinnaker user. Looking further in we identify the basic startup web page of the Spinnaker web application, index.html.

“Being able to view the web page templates and web pages is a excellent start learning about the ‘Spinnaker’ from Frank’s vantage point.

“But looking at the ‘Spinnaker’ build rules and the ‘Spinnaker’ artifacts that Frank held in his local svn version of suggests two anomalies. The first anomaly seen in his web svn shows the database content with a database backup. The second anomaly is that his web svn is the ‘e2Spn/web’ folder has an ‘extra’ sub-folder caller ‘todo\_list’. The ‘todo\_list’ folder provides a list of enhancements, issues, status, and Agile project artifacts: roadmaps, user stories, class work, and release planning.

“A point of note is that Frank seems to have been required to develop his segments of the ‘Spinnaker’ project on a Catalina storage media. He would have left ‘hidden svn files’ in each and every folder that served as breadcrumbs for his local svn archive. That is how svn configuration works. It would also be a large flag to anyone monitoring Frank development activity that intellectual property was being duplicate locally. That observation would seem to imply that Frank most likely only risked making documenting the Spinnaker application development immediately before the ‘close-out’ meeting with Catalina.

“This looks like a good place to go scratch my head for awhile and break for lunch,” muses Lionel as he closes-down for a brief lunch.

“Returning from lunch Lionel mutters to himself, “We may have found the Rosetta Stone of the ‘Spinnaker’ project. The web folder’s ‘todo\_list’ sub-folder definitively marks ‘Spinnaker’ as an Agile development project.

“To proceed we need to review that ‘early’ email with the original Releases and Sprints. Then we can compare that early ‘vision’ with what we find in the ‘todo\_list’, ‘schema’, ‘html’, and ‘templates’.

“From that mid-SEP 2020 email from [architect@anetswarf.com](mailto:architect@anetswarf.com) we know the original plan called to three releases. The first release set-up an operational environment creating a database of at least one actor and provides a virtual presence capability. The second release was to yields registration either on-line or by an eform. The second release also request an absentee ballot. Finally the third release intended to manage ballot execution and submission.

“We would think that Frank would have ‘backed-out’ of something that was beginning to seem ‘strange’ from the mid-SEP 2020 email. But that is hindsight. Plus with the COVID-19 pandemic Frank may have lapsed a bit ethically when jobs began to tighten and chaos ruled the streets.”

“The question becomes ‘what were the original individual Sprints expected to accomplish?’ The early Sprints of Release 1 seems to have been attempting to create ‘virtual’ voting citizens, to provide corresponding ‘virtual’ legends for the ‘virtual’ citizens, and finally provide the ‘virtual’ citizen the capability to generating automated keystrokes.

“The Sprints of Release 2 sought to use the USPS as a conduit for registering as a voter and requesting an absentee ballot. These Sprints would involve bring together a hardcopy of the eform, coupling it with a ‘virtual’ citizen profile and a corresponding legend. Based on experience pulling together phantom communities for military training exercises there would need to be ‘minimal support’ that had to have some physical representation. After having gone through a several of Exotic Dancer,

Solid Shield, and Robin Sage field exercises, it is feasible to believe such ‘environments’ could actually be created.

“The Sprints of Release 3 seeks to accomplish the act of voting by ‘virtual citizens’ and ‘virtual witnesses’ using computer numeric computers and the USPS conduit. Automated completion of a form is relatively easily compared to duplicating how humans ‘hand write’ signatures. But with today’s CNC automated drawing and either of three known applications available under OpenSource licenses from GITHUB.com it is feasible.”

“Jumping from that known understanding of the ‘Spinnaker’ project let’s start a deep dive into the /trunk/e4Sys/web/todo\_list/ artifacts,” Lionel thinks to himself as he begins to plow manually through the todo\_list/ artifacts.

“Just for grins let’s start with ‘todo\_list/stories’. Nothing happens in an Agile programming project without a ‘Story’ to explain why ‘something’ is included in the project. And ‘something’ is always expressed as an artifact. “

Lionel arrives at an abrupt stop at reading the first interesting artifact.

Identifier: ST00001

Name: Business Requirement

A: As a Catalina Cyber Security Agent

B: I need a safe, isolated but realistic ‘test environment’ reflective of the client’s absentee ballot election processing system

C: So that Catalina can develop appropriate cyber security and counter security measures for the State of Georgia, USA.

So the State of Georgia can prevent penetration of registered voters database

So the State of Georgia counts only valid absentee ballots

D: Acceptance Criteria: Approval of ‘super-client’ State of Georgia that all known or suspected avenues can be individually flagged for disposition and statistically tracked as a measure of processing system performance.

“If we are reading this correctly we have a collection of user stories that describe what ‘Spinnaker’ was intended be or do. At the very top in the hierarchy of the ‘Spinnaker’s’ assemblage of stories is ‘st00001\_bizz\_req’ or ‘ST00001 Business Requirement’.

“The ‘ST00001 Business Requirement’ seems to spell-out a creation of a test environment with which Catalina and the State of Georgia was going to ‘war-game’ cyber security. That explains plenty.

“First, project participants original viewed ‘Spinnaker’ as a noble, non-threatening project.

“Second, Frank would have no suspicions of something not being as advertised by Catalina when he setup his local svn.

“Third, gator and he stumbled across something that suggested a non-kosher or false-flag project aim or goal.

“Fourth, Catalina Software Engineering and its project ‘Spinnaker’ seem to have ‘skipped town’ like Frank and Stacy.”

“We are going to look a little further before raising the flag of alarm.

“Based upon what we have learned about the Catalina-Spinnaker interest in CNC and duplicating human handwriting for signatures, we must return to our ‘earlier assumptions about which repositories were related to Spinnaker. Let’s return the Frank’s local s/var/svn base web page . We originally rate freecad as ‘not relative to project’.

**Table 8.2 /var/svn/repos Collection of Repositories**

....	
eclipse_devwrk/	==> need follow-up
....	
freecad/	==> not relative to project
sloeber/	==> need follow-up

“We may find the freecad repository interesting because it could contain applications for controlling ‘human-like handwriting’ and automated form completion of hardcopy forms. The ‘sloeber’ repository could contain applications created using ‘Sloeber’ which is an enhancement of the ‘eclipse’ programming environment. Eclipse was used to create code artifacts contained in the eclipse\_devwrk/trunk/e4Spn svn project folder.

“We had better break for supper. We are having barbecue chicken leftover on sliders tonight,” recalls Lionel aloud. He closes-out the afternoon’s effort and heads to the kitchen to start supper preparations.

Returning from supper Lionel restarts the audio/video equipment. Before he proceeds to the hunt through the freecad/trunk/ repository he begins to question his direction and purpose. “If we discover any linkage with keystroke completion and automated signatures is that going to advance our understanding of the ‘Spinnaker’ and Frank-Stacy mysterious behavior and eventual demise.”

In a text message Lionel writes. “Aminu, let me know when we can have coffee again. I have a question for you that needs clarification regarding our prior project.”

Continuing to push forward for the remainder of the evening Lionel investigated the contents of the ‘freecad’ and ‘sloeber’ repositories.

From the ‘freecad repository he notes a project that unites cursive fonts, probe like behavior of a stylus, and a shallow engraving. Then there is another python program for actually controlling the LinuxCNC milling or 3D machine to accomplish minute x,y,z movements. In the ‘sloeber’ repository he found no projects that appear to be relevant to Frank’s last project.

“But for us to venture further into either of these two feasible approaches for generating human-like cursive signatures will have to wait until additional resources are available or another interested party or agency takes-up the challenge,” Lionel. “Obviously, Frank and gator were pushing the innovation envelope in their bonus money competition. It actually appears to be a collaboration more than competition.”

“Anyway it has been a full day and it is time now to close-up shop for the night,” exclaims Lionel. “I will just have to wait on Aminu to respond to the setup the coffee meeting.” Lionel turns-off the audio/video equipment and lights as he leaves the workspace/office for the evening.