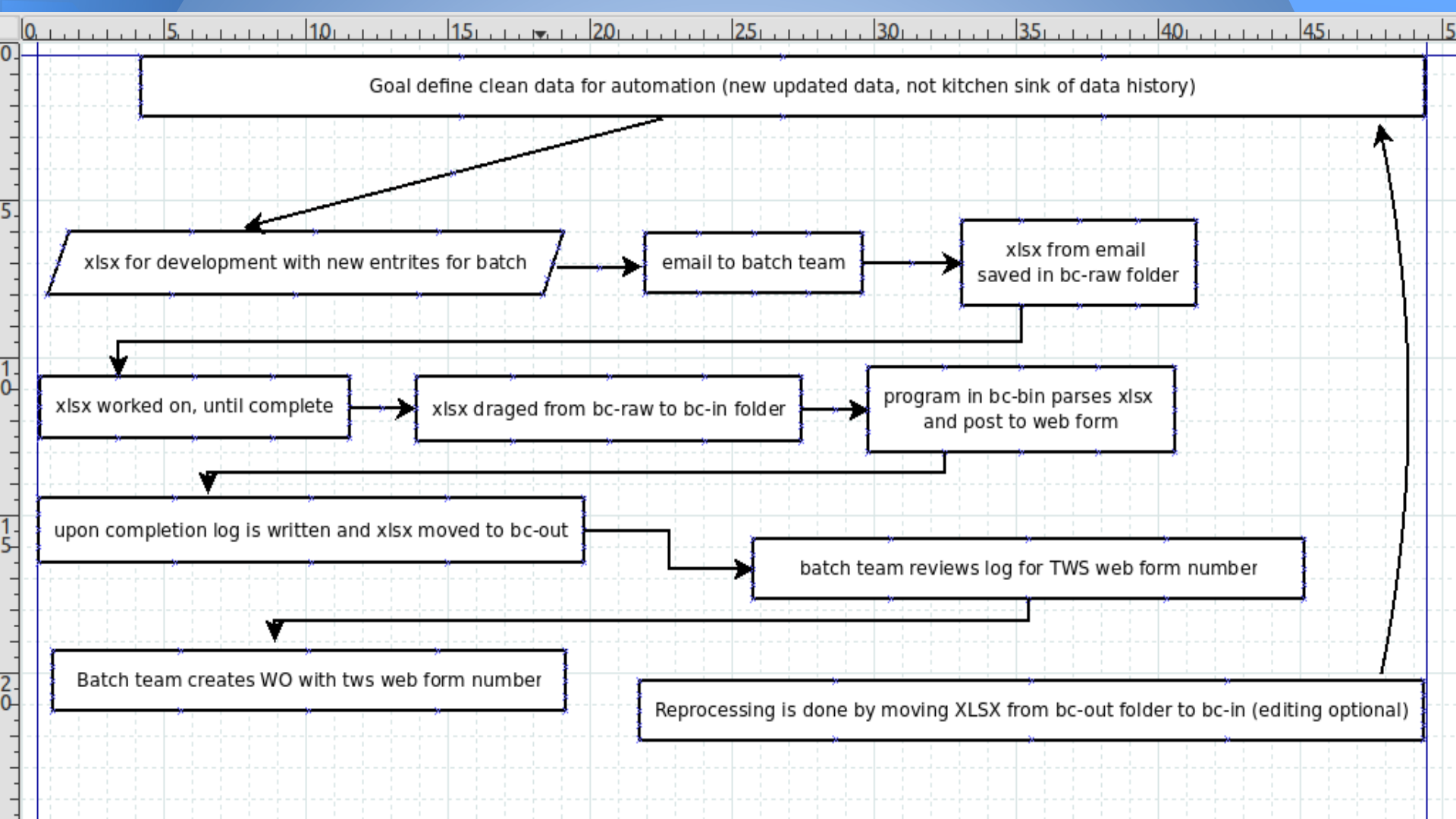
Web automation project:

Tapestry Release Management


Goal define clean data for automation (new updated data, not kitchen sink of data history)
xlsx for development with new entrites for batch
email to batch team
xlsx from email saved in bc-raw folder
xlsx worked on, until complete
xlsx draged from bc-raw to bc-in folder
program in bc-bin parses xlsx and post to web form
upon completion log is written and xlsx moved to bc-out
batch team reviews log for TWS web form number
Batch team creates WO with tws web form number
Reprocessing is done by moving XLSX from bc-out folder to bc-in (editing optional)

Requirements:

1) To use the TWS batch web form, as it is the only way to handle errors from IBM and is the only method for auditing.
2) To quickly get batch data into a supported browser in Windows 7 workstation environment.
3) Does not have to be KP wide solution, it is solution only for Tapestry batch.(keep it simple, as eyes will verify on both ends of the process)
4) Spread sheet must be able to call process that posts data to the web page.
5) Size and implementation speed are important as Batch coordinators workstation are window 7 32bit with 4GB of memory and standard process take over 90% of memory and 30% of cpu on these small workstations.
6) Minimum modification to KP secure workstations for contractors.
7) Simple distribution and integrity which reduces support.(stretch goal would be single file).

Environment changes:

New Browser install on Batch coordinators workstations.
Firefox browser is support and available from KP IT-help.
Firefox should be require because of two features.
1) Speed and compatibility
2) Only browser to support native Selenium webdriver without additional binary shims (see W3 specifications).
3) Skip IE and chrome because they require an additional layer of a binary translator, which is not as fast or reliable. Also these require network port unlocking (KP workstations currently have network restrictions)

Technical option 1:

Free OCR scraper to drive the browser application.

+ Able to automate hidden fields (Activex,Applets,Ajax,silverlight,ect.)

- Larger footprint, Sikuli/sikuli-slides jar (56MB) plus scripting runtime(126MB). Uses PNG images, managed locally

- Slides data interface currently is with command line parameters.

- User can interfere with automation by using the keyboard and mouse during automation.

+ No network changes requires

Technical option 2:

Free Selenium webdriver to drive the browser application.

+ Tried and true, mainstream support and use in the industry.

+ using the web browser DOM to id the html/CSS and then send the data to the web page, bypass interruption from users keyboard and mouse.

+ It is possible to distribute a single small executable, which is way more robust.

+ Smaller foot print as single executable (this eliminates scripting languages like python and ruby, which have to also be updated (maintained) and consume disk space.

+ Multiple browser, and programming language support

+ Local direct browser mode for Firefox, which allows security compliance. (server mode supports top three browser, but requires network server an ports to be active)

Technical option 3:

Free browser macro script to drive the browser application.

+ built into Firefox (uses Selenium).

+ Small footprint.

+ Single VBA Macro in spreadsheet, spreadsheet would need to be locked (need more research)

- User can interfere with automation by using the keyboard and mouse during automation.

+ May be able to have a spreadsheet macro to create a JavaScript data file and start the browser, JavaScript is modifiable on workstation, leading to support issues. (need more research)

Technical option 4:

QTP usally the first instance is $100K, +seats. program that allows interaction with COM objects using VB.

- Much older technique and often problematic, might need QA test developer to develop script
- Often require custom scripts and support from HP, slowing development.
+ Already used in KP for testing, KP has site liscense.
+ Can create executable for project (recommended as gui fails 50% of time).
- Requires full install of QTP framework to be for exe to run.

Technical Option #5

Free Mouse and keyboard com scripting tools (best use for command line only automation). Number of tools available (autoit, autohotkey, javauto, pyahk, pywinauto)

- Much older technique and often problematic
+ Able to automate hidden fields (Activex,Applets,Ajax,silverlight,ect.)

- Blind automation, via coordinates
- Affected by screen size and position changes
- Affected by mouse, keyboard and message event changes
- Difficult to keep running unless hosted in quarantine clean VM

Tasks:

Completed:

Identify the web object (CSS and HTML).

Identify the XLS objects (fields).

Research automation methods.

Identified workflow process.

To do:

Get Web batch test account.

Prototype with CSV file.

Enhance spreadsheet with data validation and simplification (pull downs of predefined values), macros to save as csv and invoke the automated web posting program (XLS wizard, Jayme and Alicia have volunteered to help with the VBA macro).

Summary:

+ I Have familiarity with options 1,2,3,4 this important as development does not available resources or familiarity with 1,2,3. ALM does have familiarity with option 4.
+ I Have implemented similar automation before using options 1,2,3.
show Selenium local option #2 or Selenium IDE macro #3 as most viable.
+ Technical options scorecard

#2= +6 ,-0;	#3= +4,-1;	#4= +2,-2;
#1 = +2,-3;	#5=+1,-5	

Schedule (given resources for accounts, testing)
Prototype 2 week, Testing 1 week (given resources and access approvals)

References:

Short example fromhttp://forums.techguy.org/business-applications/822451-vba-macro-opens-command-prompt.html

So the XLS macro would call option #3 like:

strPrgm="c:\data\bin\BC-loader.exe”

strParam1="c:\data\save-csv\tws.csv"

……...

http://www.thespreadsheetguru.com/getting-started-with-vba

http://www.thespreadsheetguru.com/blog/

https://dvcs.w3.org/hg/webdriver/raw-file/default/webdriver-spec.html

http://www.sikulix.com/

http://slides.sikuli.org/

http://software-testing-tutorials-automation.blogspot.com/2013/07/parameterization-in-selenium-ide.html

https://github.com/tourdedave/selenium-benchmark

http://elementalselenium.com/

http://software-testing-tutorials-automation.blogspot.com/2013/07/parameterization-in-selenium-ide

http://javauto.org/.html

http://www.sciencedirect.com/science/article/pii/S2213133714000687

Addendum

1. Jruby (more stable version ruby, with threading) selenium works but has to have complete install of support. More difficult to make a native binary.
2. Implementations that require selenium host server need open network port on local machine (default port is blocked and can be changed)
3. Ironpython and selenium has issues with setups
4. Opted out of Sikuli-slides since it does not read XLXS
5. Opted out of sikulix, since install require admin install, project still in development
6. Python selenium single threaded (gil problem) , py2exe for packaging, allow distribution with out install (drawback is a very large non compiled package). Also works with openpyxl to parse spreadsheets 2x faster than jruby
7. Options to compile python: Pysco and shedskin are obsolete, hope is Linux only, pypy still has gil issues, pyston is still being developed.
8. Go has multiple implementations selenium webdriver (version 2 of selenium language) , but go-selenium is the most complete, with fast and small binary.
9. Using selenese (version 1 of selenium language from the IDE) from the command line is not option due to not being able to parse spreadsheets and is not compiled.