WHY YOU SHOULD HIRE ME:

I am enthusiastic about improving business with modern information systems. I also bring many years of experience with project management, application development, systems engineering, which allow efficient synergy with various stakeholders.

EXPERIENCE:

Directly relevant: I have provide technical strategy to companies to bring in new tools and techniques. At Boeing I was the innovator bring in C++ to mini and mainframe computers. At ADP I was the innovator bring in KSH, Perl, Windows, Virtual environments, Automated testing. At Rentrak, I was the innovator bringing in continuous integration and continuous delivery with new tools, processes and procedures. At Kaiser, I was the innovator bring in automated tracking build and release system and tools.

Systems and application engineering background: I worked as a systems software engineer in operating systems, installs of internal and external software products. I worked as a software developer in creating custom applications. I worked as a senior application lead to created search engines and websites. I have done legacy application and platform migrations, along with new platform and application development.

Research and analysis experience: At ADP, I wrote RFP's for third party products that integrated into our projects. These RFP's require unlimited licensing and rebranding of the third party products, along with customization to meet our business requirements. I have also worked on RFP's for service providers for cloud VM and testing resources at Rentrak. As a software engineer, I have always researched new tools and process, which would reduce development and maintenance costs, along with improving time to market.

Requirements gathering: As a project manager, I have found business requirement often need revision to add quantifiable goals and measurable metrics. I have worked with business and marketing team to write requirements as precise single steps, rather than vague requirements, which better manages the programs being developed.

Broad understanding of IT delivery: I worked using the SDLC waterfall method of development. I have also work and had my teams using lean, XP, Agile rapid development for product development. I work to careful balance the business needs of quality, time to market and functionality with development schedules and resources.

Project management: I have done project management for several years and project at ADP for a core set of custom services that provide the base for all of ADP dealer services applications. These services have separate life cycles in specific environments that had to be maintain and supported. There were client test environments that had to be setup and monitored for external and internal customers. Along with the deployment of these services which were distributed through several methods of on-site, and cloud base environments. In addition, at later projects at Huron, Rentrak, and Kaiser I managed similar projects that supported integration, test and deployment services. I obtained a PMP certificate in 2006 from Cadence and I currently am working to refresh my PMP certificate and new certification as a Scrum master.

(The following are helpful, not required. Please not your experience, if any)

SAP: I have had experience with Service Oriented Architecture products like SAP, in the areas of CRM, ETL, ERP, and Data warehouses. I have also created custom enterprise real time tracking systems at Boeing, ADP, Rentrak and Kaiser for status and logistics. Comprising of tracking systems that allow queries of where the product deliverables were in the process pipeline. This saved the companies hundreds of thousand dollars over previous antiquated systems. I believe it best to understand the business workflow and vendor's workflow in order to provide the highest ROI.

Linux: Worked in UNIX at Boeing. Worked as Systems software engineer in UNIX/Linux at ADP creating server products and install. Worked with QA integration, automation for Linux APIs' and Web products at ADP.

Worked in Linux at Rentrak for automation, build, integration, release, and testing. Worked with AIX at Kaiser, which is a UNIX like OS.

Virtualization: I have use Microsoft Hyper visor, VMWare and Virtual box. : I was the innovator who brought VM systems into ADP to perform QA integration and testing for both servers and clients. I have continued to use VM systems at subsequent jobs as they provide a rapid and consistent platform, which is cost effective.

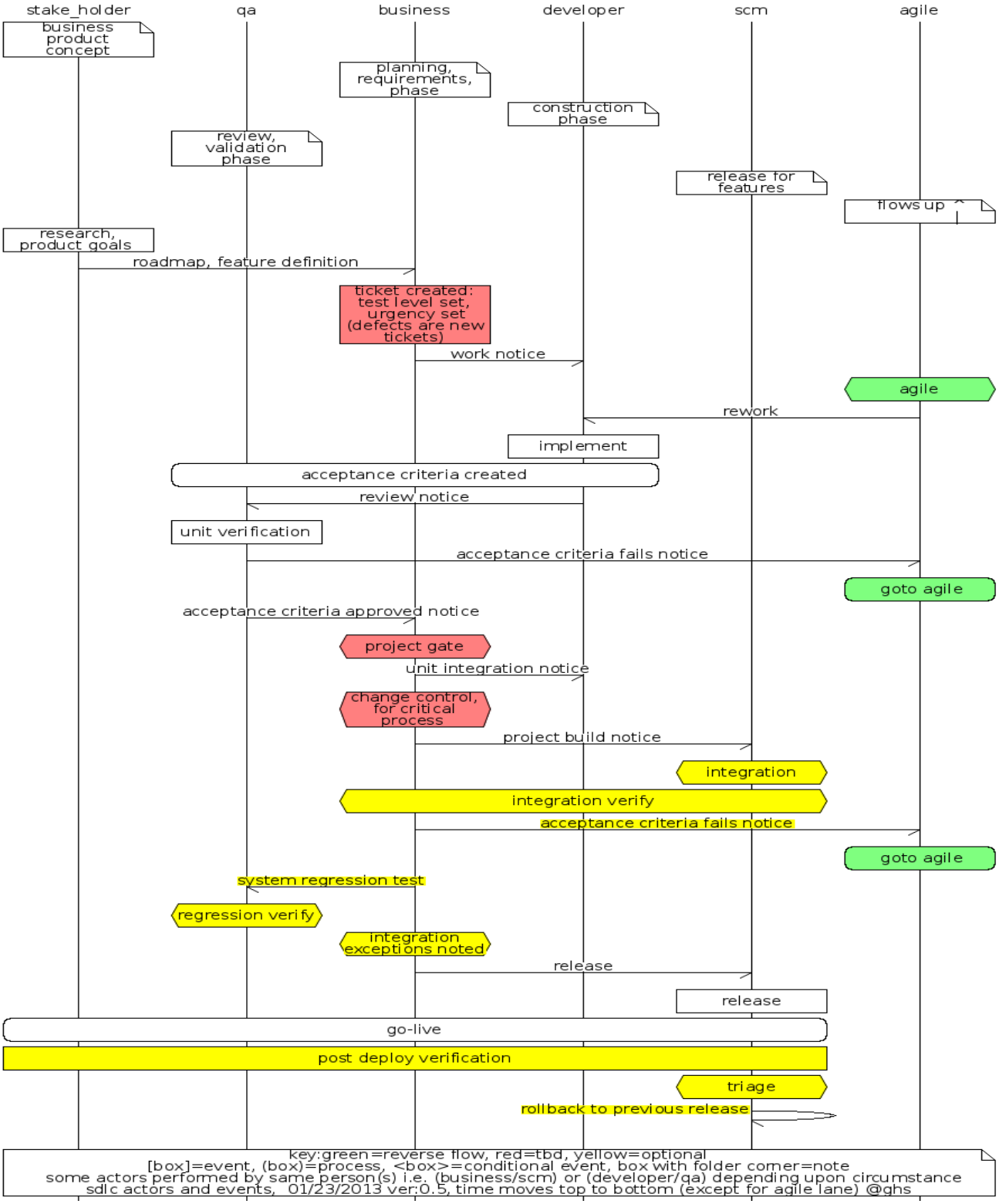
Cloud environment: I have managed deployments of cloud services, with projects at Huron, Rentrak, and Kaiser. My research on cloud applications with IaaS is to start with mapping of the services, data, and network events to determines where services are and granularity, clusters, databases, caches, multiple access(desktop, phone), shared resources are to be store and where process are to execute in cloud data centers. This determines which components to distribute, as the down side is the management of distributed systems. Making sure the cloud components have priority ranking, monitoring, performance, redundancy, backup and security. At both ADP and Rentrak, the cloud projects required having fail over systems for critical VM systems and databases. At Kaiser it was the same with the addition of having service level agreements, which differed by the priority of the service. The determination of the priority, quality and recover of the cloud service requires taking into account the relationship each component of the cloud service and the service provider's reliability, support, scalability and security. My team used PaaS in a pilot proto type of Google application engine, where we had a number of issues with scalability, performance and concerns on vendor restrictions and long term support

VM Ware: I have spec hardware and service for ESX, along with managed VM Ware instances.

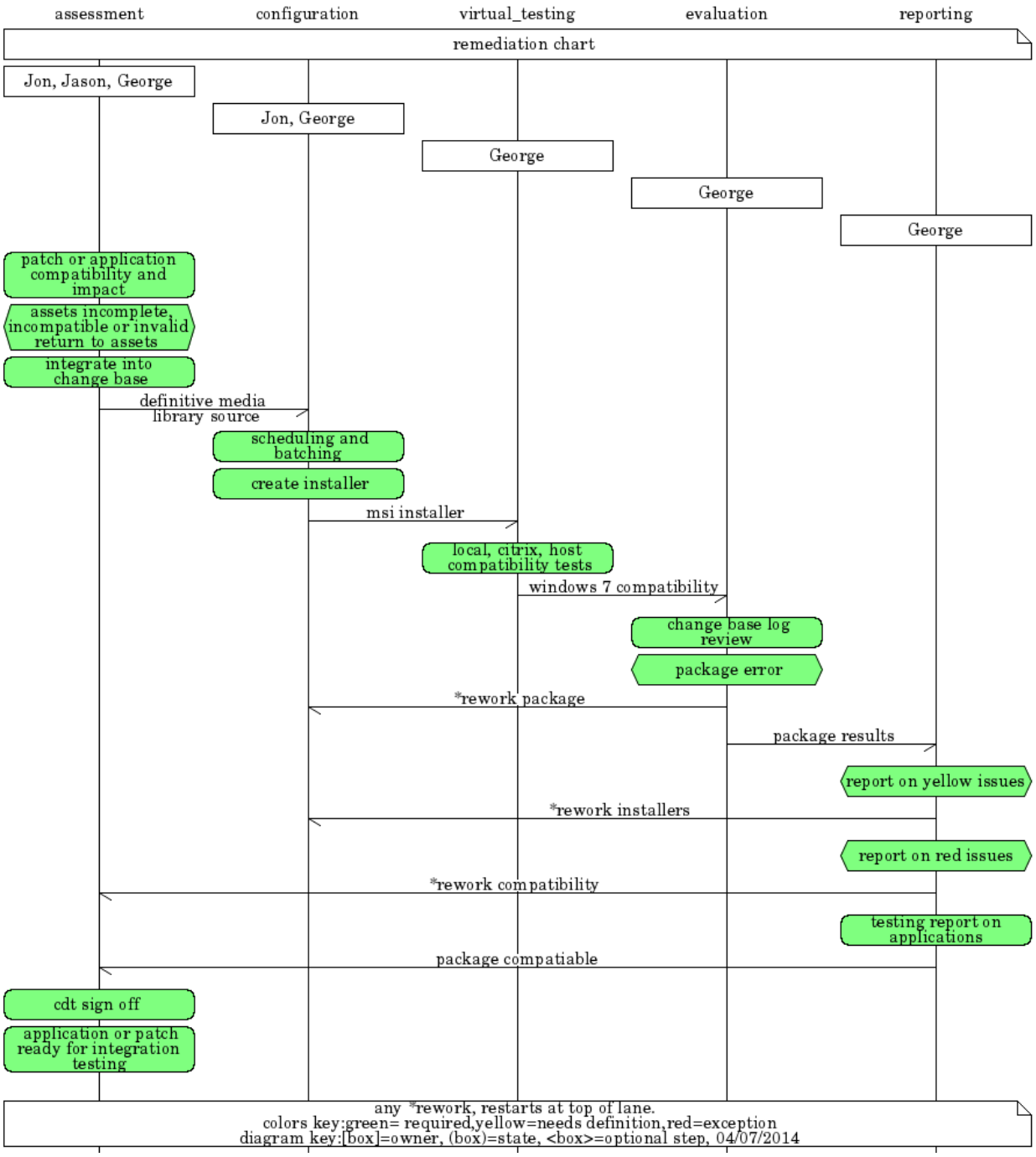
Enterprise architecture: My experience in designing system and platforms, I have designed platforms at ADP that user system services. These were both local and cloud platforms. This involved; hardware, software, service level agreements and support environments.

Solutions design:

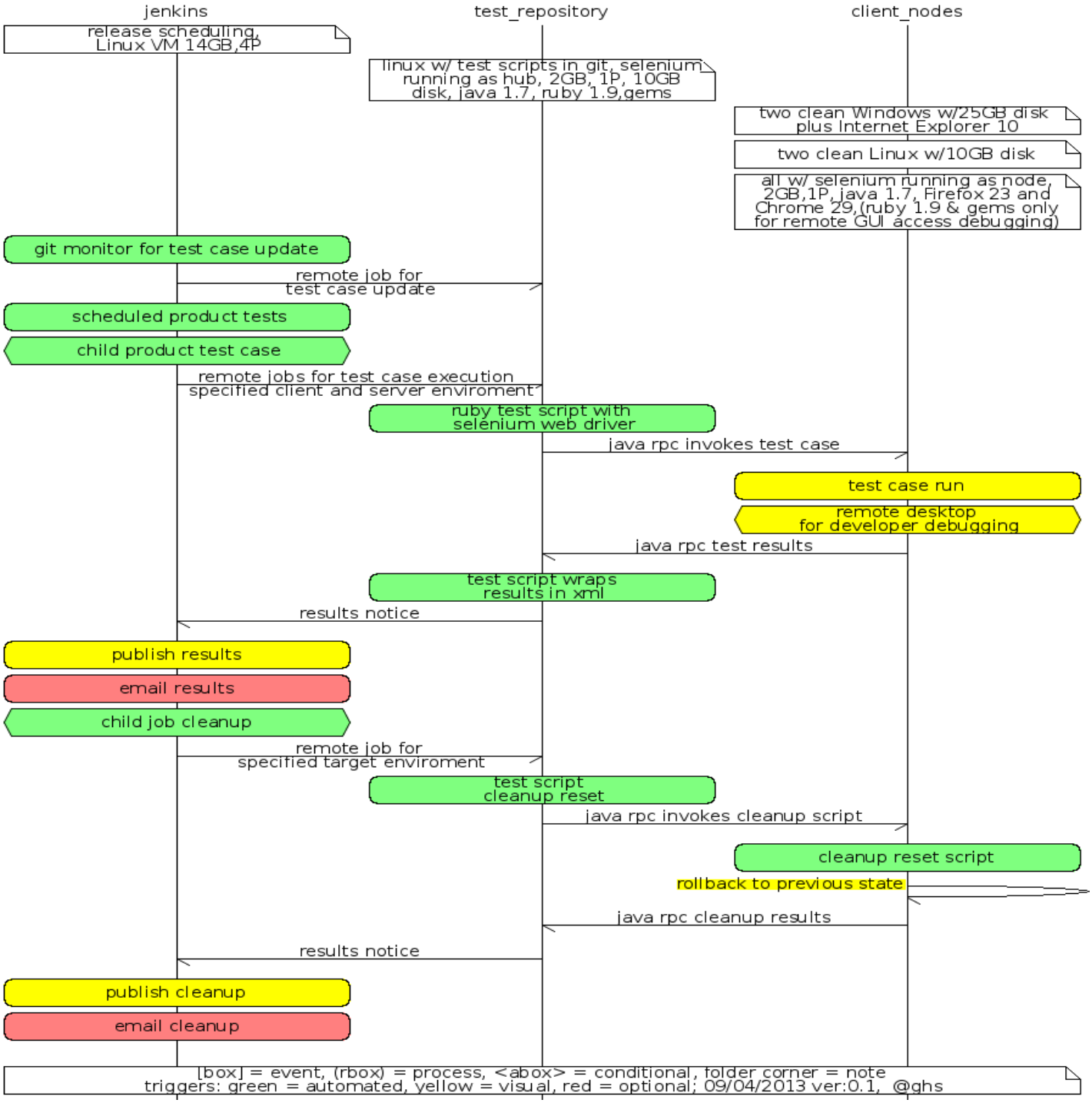
- My experience is to create and gather the information and clearly document the process, before changing the process. At Rentrak, I implemented a conversion from a legacy system to a modern implementation. Below is a sequence diagram of the process workflow, which defines the system (actors), events and message transitions for replacement system.


stake_holder
qa
business
developer
scm
agile
business product concept
planning, requirements, phase
construction phase
review, validation phase
release for features
flows up
research, product goals
roadmap, feature definition
ticket created: test level set, urgency set (defects are new tickets)
work notice
agile
rework
implement
acceptance criteria created
review notice
unit verification
acceptance criteria fails notice
goto agile
acceptance criteria approved notice
project gate
unit integration notice
change control, for critical process
project build notice
integration
integration verify
acceptance criteria fails notice
goto agile
system regression test
regression verify
integration exceptions noted
release
release
go-live
post deploy verification
triage
rollback to previous release
key:green=reverse flow, red=tbd, yellow=optional
[box]=event, (box)=process, <box>=conditional event, box with folder corner=note
some actors performed by same person(s) i.e. (business/scm) or (developer/qa) depending upon circumstance
sdlc actors and events, 01/23/2013 ver:0.5, time moves top to bottom (except for agile lane) @ghs

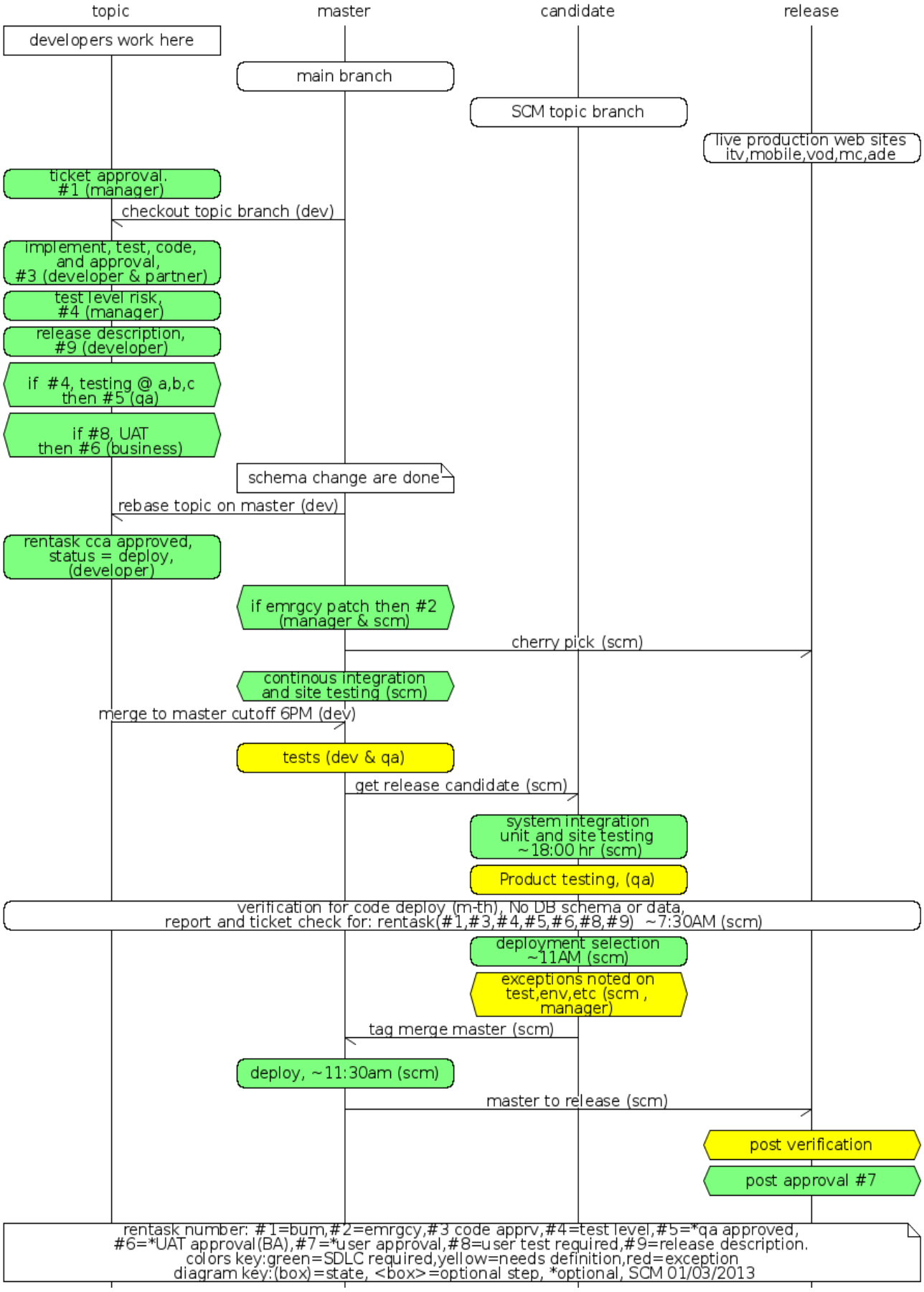
- Here is another example of a sequence chart on the conversion from a legacy system to a modern implementation for testing, I implemented at Salem Heath.



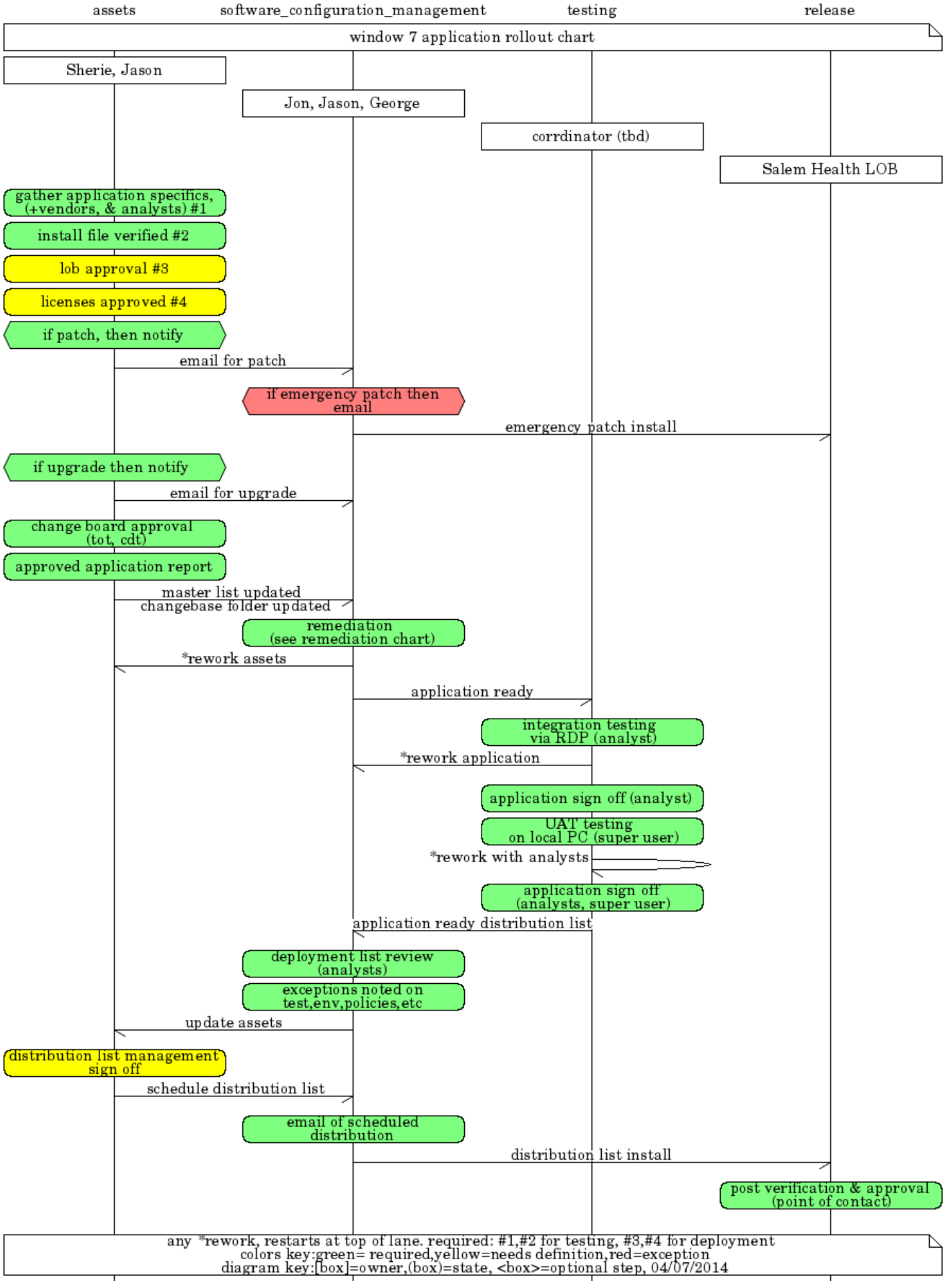
- Here is example of my conversion from a legacy system to a modern implementation for integration, I implemented at Rentrak.


jenkins
test_repository
client_nodes
release scheduling, Linux VM 14GB,4P
linux w/ test scripts in git, selenium running as hub, 2GB, 1P, 10GB disk, java 1.7, ruby 1.9,gems
two clean Windows w/25GB disk plus Internet Explorer 10
two clean Linux w/10GB disk
all w/ selenium running as node, 2GB,1P, java 1.7, Firefox 23 and Chrome 29,(ruby 1.9 & gems only for remote GUI access debugging)
git monitor for test case update
remote job for test case update
scheduled product tests
child product test case
remote jobs for test case execution specified client and server enviroment
ruby test script with selenium web driver
java rpc invokes test case
test case run
remote desktop for developer debugging
java rpc test results
test script wraps results in xml
results notice
publish results
email results
child job cleanup
remote job for specified target enviroment
test script cleanup reset
java rpc invokes cleanup script
cleanup reset script
rollback to previous state
java rpc cleanup results
results notice
publish cleanup
email cleanup
[box] = event, (rbox) = process, <abox> = conditional, folder corner = note
triggers: green = automated, yellow = visual, red = optional; 09/04/2013 ver:0.1, @ghs

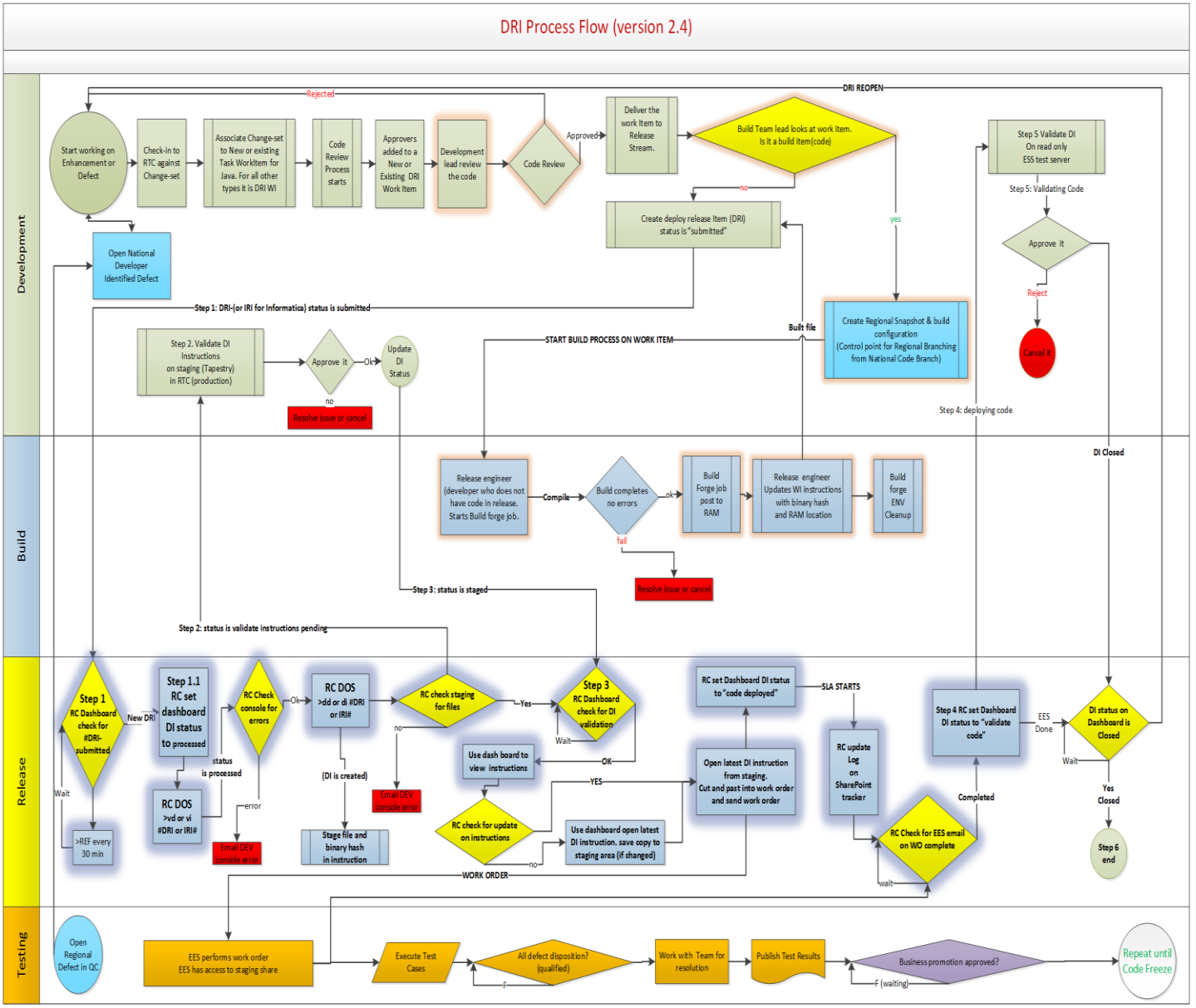
- Here is an example my conversion from a legacy system to a modern implementation for source control, I implemented at Rentrak.



- Here is another example of a sequence chart on the conversion from a legacy XP to Windows 7, which I implemented at Salem Health.



- Here is an example of my conversion from a legacy system to a modern implementation for a national build and release, I implemented at Kaiser.



My brief approach is to start with a schedule, budget and scope, which the business requires, for the replacement systems. This provides the guidelines for the implementation of the new systems. Frist is to do a data analyst and data model of legacy data. The current quality of the data is very important. You need to audit the quality of the data to ensure input to the new system is valid. Resolve questions like if the data using different codes for the same component or object. Map out the data flow of the legacy systems to show vendors and versions of data sources. Mapping legacy systems interfaces, in terms of location, dependencies, service level or legal agreements, and infrastructure to or from other systems. Once the legacy system is mapped for existing functionality, the planning for the new system can begin with writing new requirements and the schedule using the business process that apply in the business environment (SDLC/ITIL). Making sure to include migrations plans for handling data loading, synchronizing, verification and sunset of legacy systems.

Solutions Delivery: Managed the continuous product builds, integration delivery/release at Huron, Rentrak and Kaiser. Created continuous delivery and continuous integration for builds and testing at Rentrak. Created continuous integration testing at ADP.

Technical lead experience: I have had hands-on experience with managing a team of IT professionals including setting expectations and measuring performance. I have had teams of software developers, QA, and business analysts. The business environment varied from formal waterfall long-term development to rapid agile

The size of the direct team was 5 to 7, the size when I managed integration projects the team size was up to 56 in a matrix reporting structure.