BACK TO BASICS

Child Welfare Information Technology Systems Managers and Staff Webinar Series

Impact of Development Methodologies Software Frameworks and Platform Choices on Release Management

Thursday, August 29, 2013
2:30 – 3:30 p.m. EDT

Joyce Rose, ICF
Tom Kine, MN
Iroabuchi Arum, OH
Fred Crawley, OH
Thomas Hammons, OH

Presented by ICF International under contract with the U.S. Department of Health and Human Services Administration for Children and Families, Children's Bureau

Back to Basics Series

- One webinar per m onth between April and September, 2013
- Target audience
  - Child welfare IT systems managers and staff
  - New and experienced
- Recorded webinars
  - http://www.acf.hhs.gov/programs/cb/research-data-technology/state-tribal-info-systems/training

Back to Basics Series (continued)

- Webinar 1 : What is a Child Welfare Information Technology Systems Manager (A pril)
- Webinar 2 : The Child Welfare Information Technology System Project Lifecycle, Part 1 (May)
- Webinar 3 : The Child Welfare Information Technology System Project Lifecycle, Part 2 (June)
- Webinar 4 : Common Pitfalls and How to Avoid Them (July)
- Webinars 5: The Impact of Development Methodologies, Software Frameworks and Platform Choices on Release Management (August)
- Webinar 6: To be determined (September)

Participating in Today's Webinar

- Questions and comments by telephone
- Questions via chat
- If you have additional questions, please contact Joyce Rose after the webinar

Joyce@kassets.com

Attendee Poll

Who is attending today's webinar?

- State Child Welfare Information System (CWIS) Project Manager
- State Child Welfare Information System (CWIS) Program Manager
- State Child Welfare Information System (CWIS) Technical Manager
- State Child Welfare Information System (CWIS) Project Staff
- ACF/Children's Bureau Personnel

Today's Agenda

- Format
- Introductions
- State participant discussion
- Attendee Q & A
- Wrap up

Introductions

- Tom Kine, MN
- Fred Crawley, OH
- Thomas Hammons, OH
- Iroabuchi Arum, OH
- Joyce Rose, ICF International

State Presentation: Minnesota

State Presentation: Minnesota

- State background
- Development methodology
- Software framework
- System platform
- Impacts upon release management
- Lessons learned
- If we had to do it all over again, we would (fill in the blank)

State Background

Child Protection
Foster Care
Adoption
Children's Mental Health
Other Child Welfare Programs
Adult Maltreatment Reporting
Waiver Claiming
Other Adult Services

State Background (cont.)

Minnesota is state-supervised and county-administered

The Minnesota SACWIS (Social Services Information System – SSIS) is used by 80 individual counties, 7 additional counties organized into 2 regions, and 2 Indian tribes

Over 6,000 agency social workers, case aides and DHS staff use SSIS to track cases involving 272,000 individuals annually

State Background (cont.)

SSIS started planning in 1995; no statewide social services system existed

Years of negotiation with counties

Counties wanted to keep their data separate from each other

Development Methodology

Waterfall for fiscal-related development

Rapid Application Development (RAD) for social-services-related development

Agile for managing bug fixes and minor enhancement requests

Development Methodology (cont.)

- Waterfall for fiscal-related development
  - Pros
    - Good with well-defines and understood requirements
    - High degree of control for financial systems considerations
    - Allows creation of detailed testing plans based on the design documentation
  - Cons
    - Rigid, hard to adjust for changes in requirements

Development Methodology (cont.)

- RAD for social-services-related development
  - Pros
    - Good for development focuses on user-interaction
    - Fast iterations focused on incremental improvements
  - Cons
    - Testing can be a challenge

Development Methodology (cont.)

Agile for managing bug fixes and minor enhancement requests

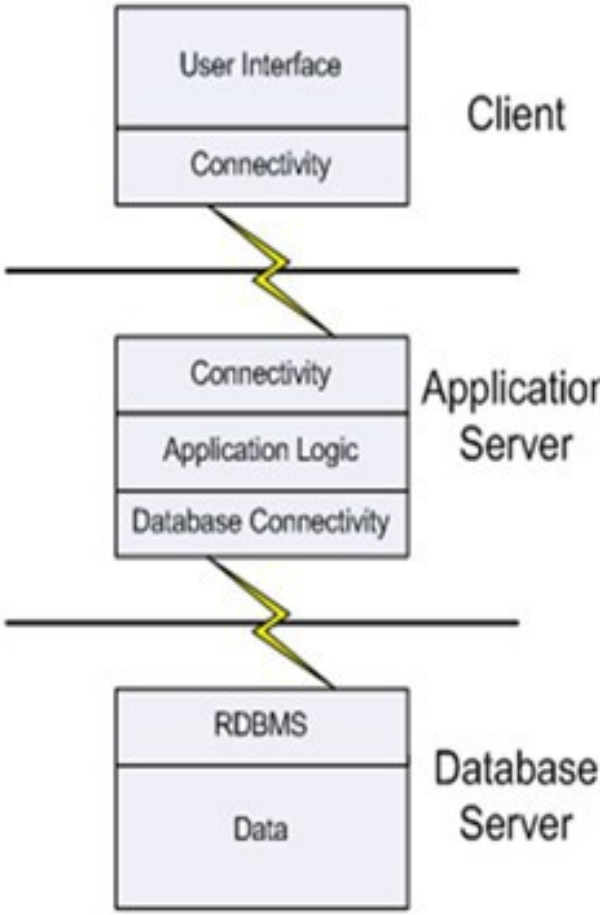
- Pros
  - Flexible as compared to waterfall
  - Quick turn-around through iterations
  - QA through-out process – fewer defects
- Cons
  - Hard to get/maintain documentation

Software Framework

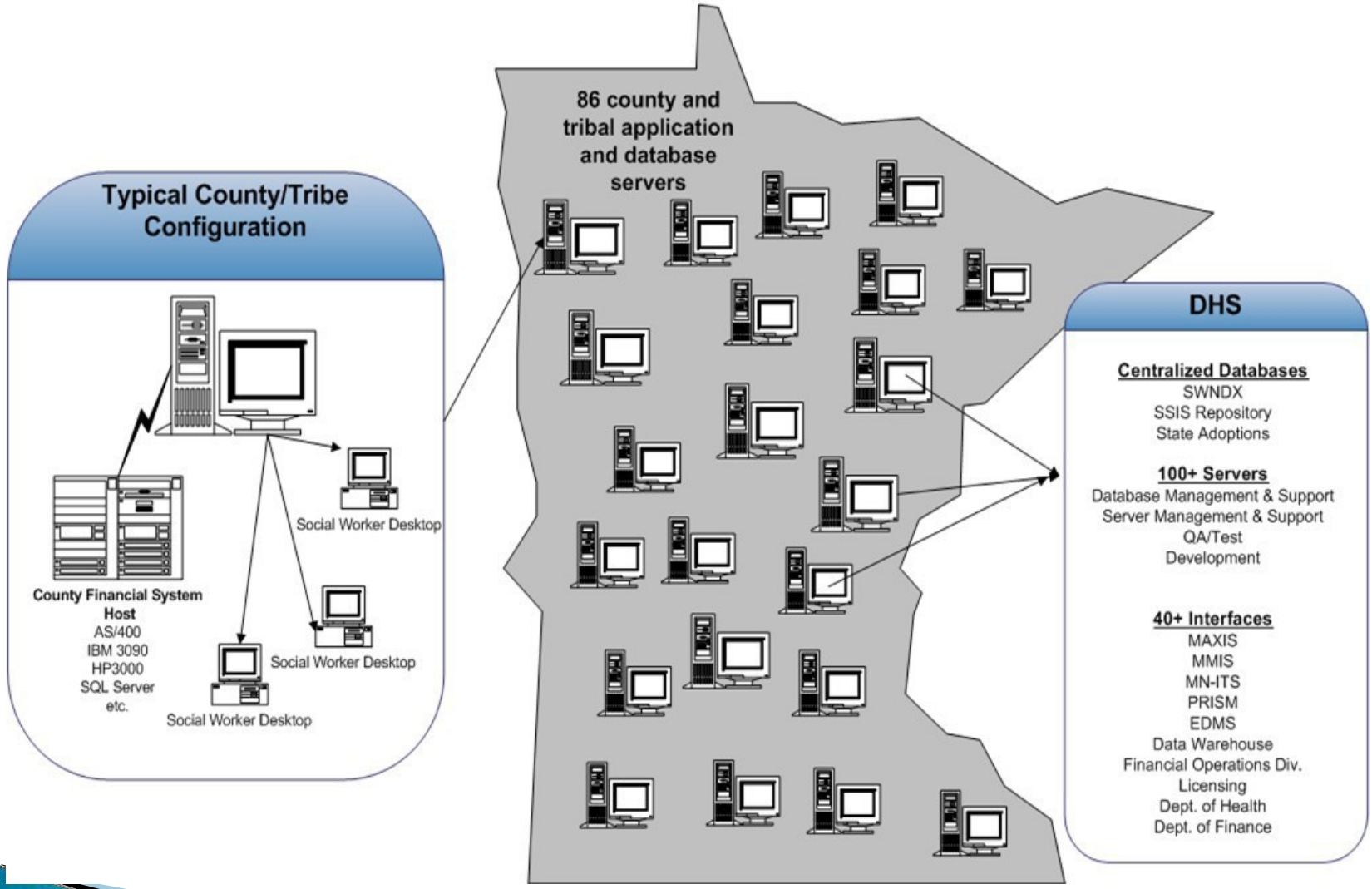
Windows Application (Delphi)

Windows Server Application (Delphi)

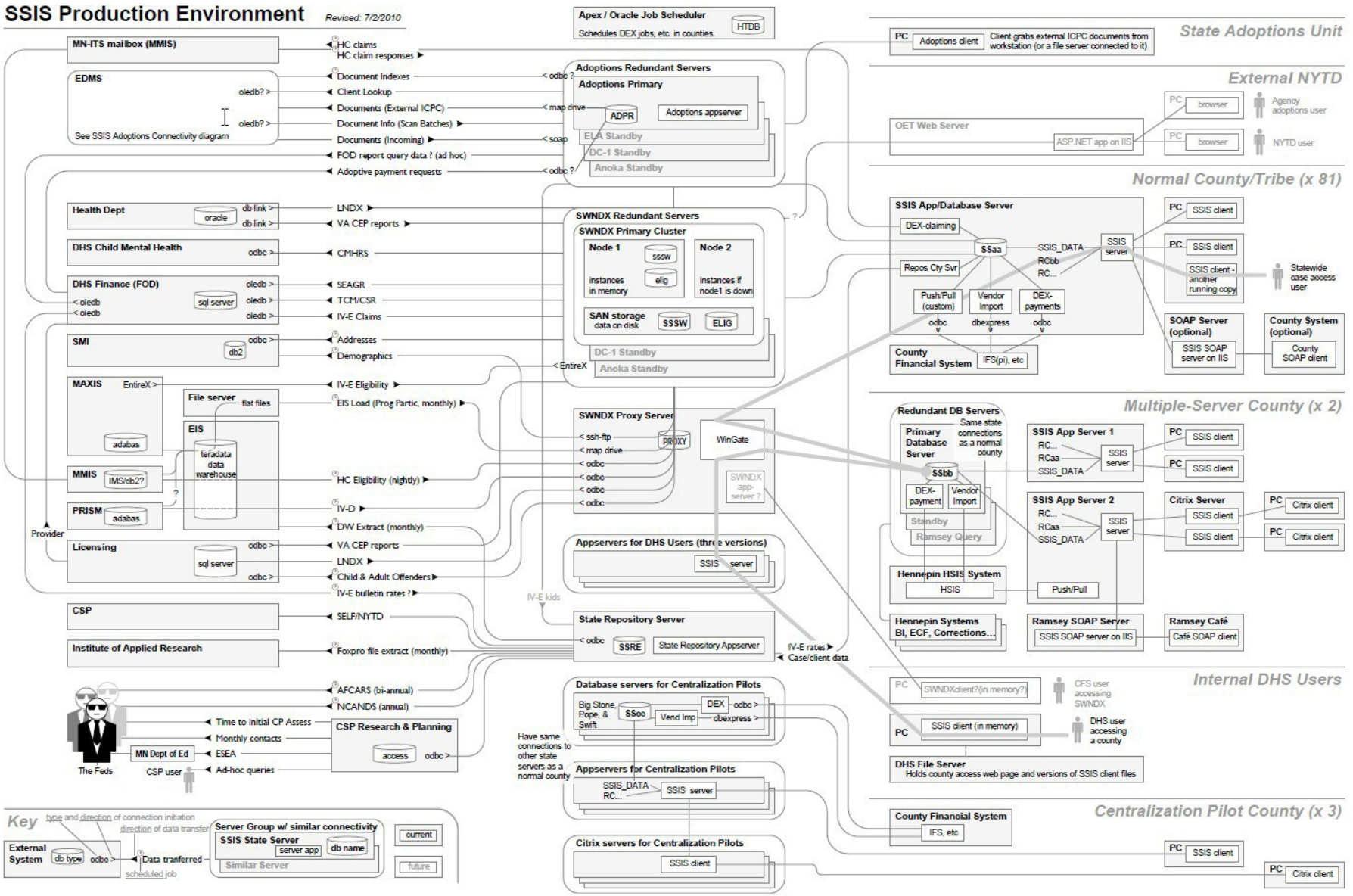
Oracle Database



Software Framework (cont.)



Software Framework (cont.)



Impacts on Release Management

- Distributing software to 6,000+ desktops is a big deal
- Historically, about 3 releases every 2 years
- SSIS uses a custom-written application launcher
  - Launcher runs, checks server for updates, if update found, patches or replaces existing .exe, launches application
  - Patch technology minimizes impact of pulling down entire .exe over county networks

Impacts on Release Management (cont.)

Agile project management has had a huge impact on release management

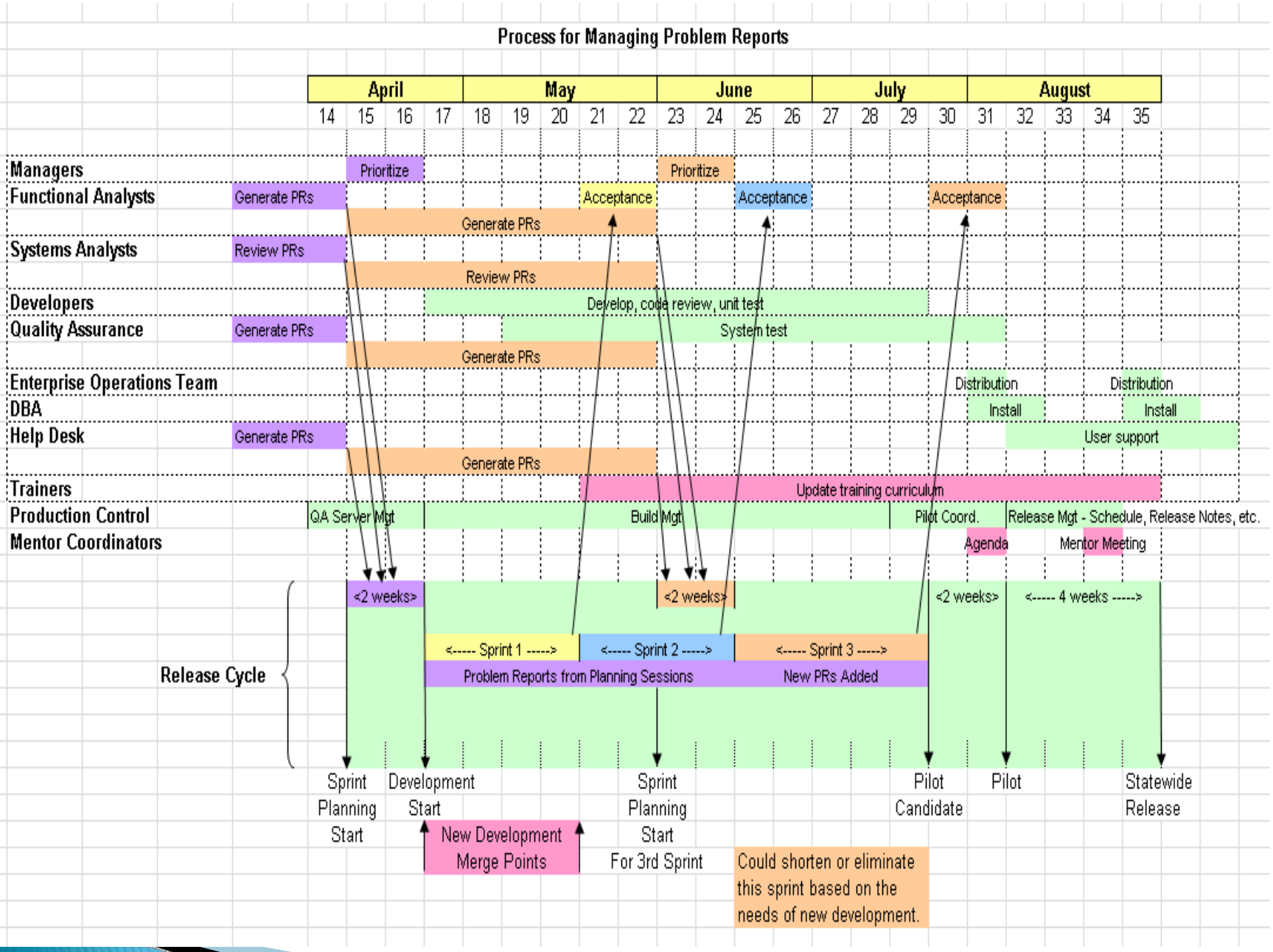
Converted to a quarterly release cycle for bug fixes and minor enhancements

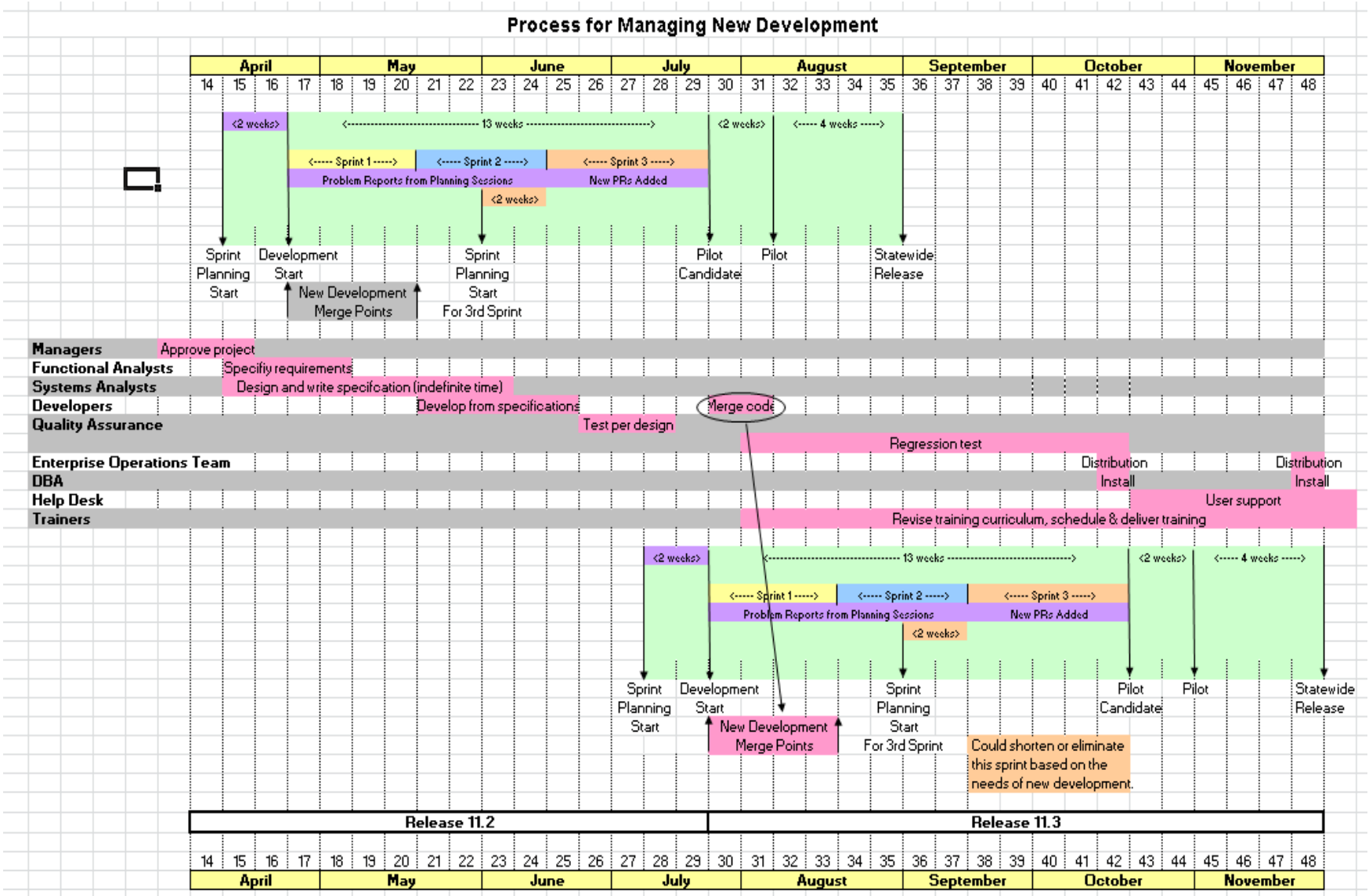
Develop new functionality in it's own branch and merge in when ready

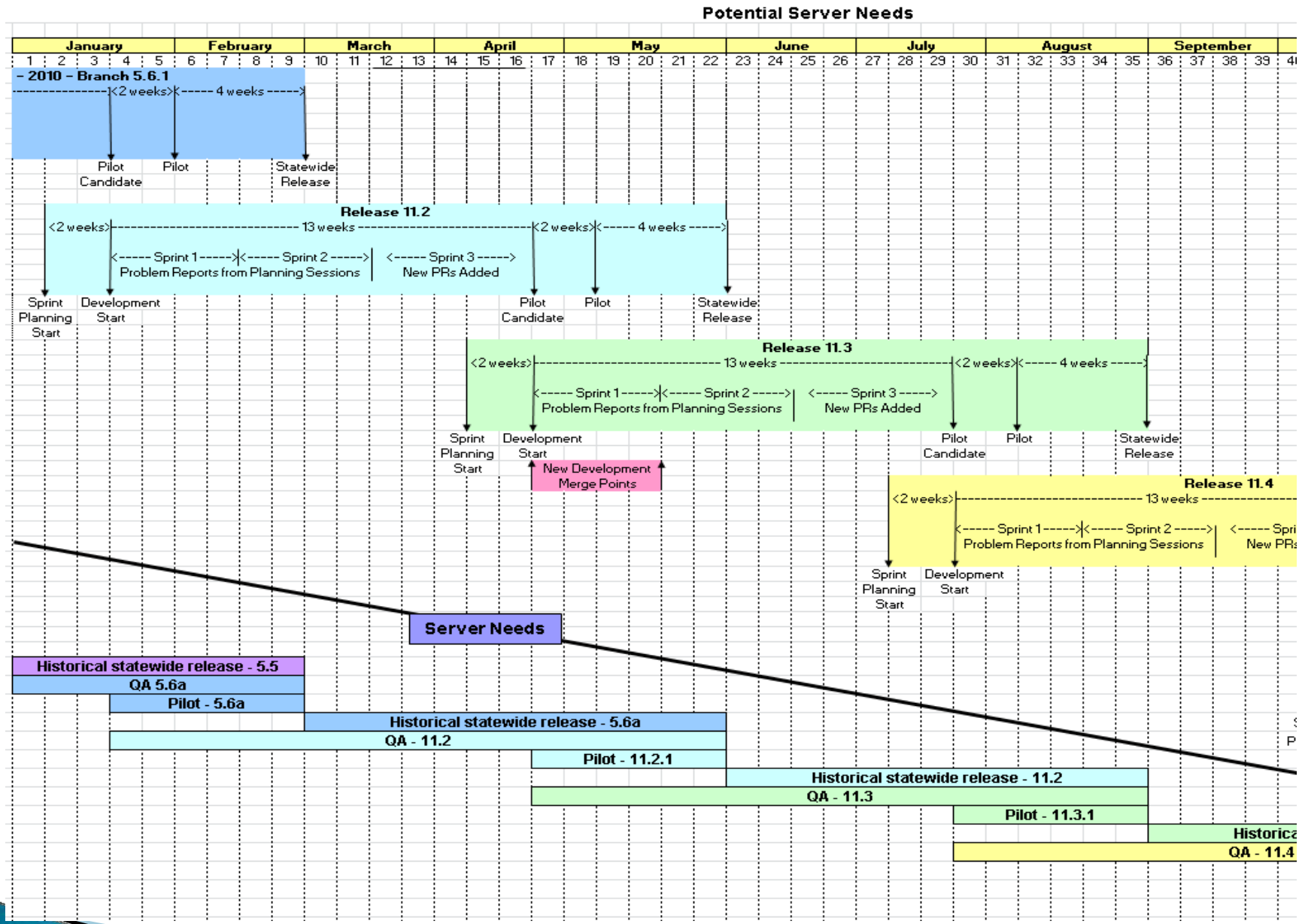
- Releases were delayed in the past due to unpredictability of new development
- The later they were, the more behind we got – bug fixes continued to be pushed into release

Agile Project Management

- Background on its introductions into SSIS
- 2009 – Attended state IT Symposium with SSIS GUI Development Supervisor
- Started Agile planning
  - Joined state self-study group for Scrum-master
  - Started educating project staff on Agile
  - Developed plan to implement Agile for bug reports
- 2011 – Started using Agile
- 2013 – 3rd or 4th release in a row delivered on time


Process for Managing Problem Reports
April
May
June
July
August
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
Managers
Functional Analysts
Systems Analysts
Developers
Quality Assurance
Enterprise Operations Team
DBA
Help Desk
Trainers
Production Control
Mentor Coordinators
Prioritize
Prioritize
Generate PRs
Acceptance
Acceptance
Acceptance
Generate PRs
Review PRs
Review PRs
Develop, code review, unit test
Generate PRs
System test
Generate PRs
Distribution
Distribution
Install
Install
Generate PRs
User support
Generate PRs
Update training curriculum
QA Server Mgt
Build Mgt
Pilot Coord.
Release Mgt - Schedule, Release Notes, etc.
Agenda
Mentor Meeting
Release Cycle
<2 weeks>
<2 weeks>
<2 weeks>
<----- 4 weeks ----->
<----- Sprint 1 ----->
<----- Sprint 2 ----->
<----- Sprint 3 ----->
Problem Reports from Planning Sessions
New PRs Added
Sprint Planning Start
Development Start
Sprint Planning Start For 3rd Sprint
Pilot Candidate
Pilot
Statewide Release
New Development Merge Points
Could shorten or eliminate this sprint based on the needs of new development.

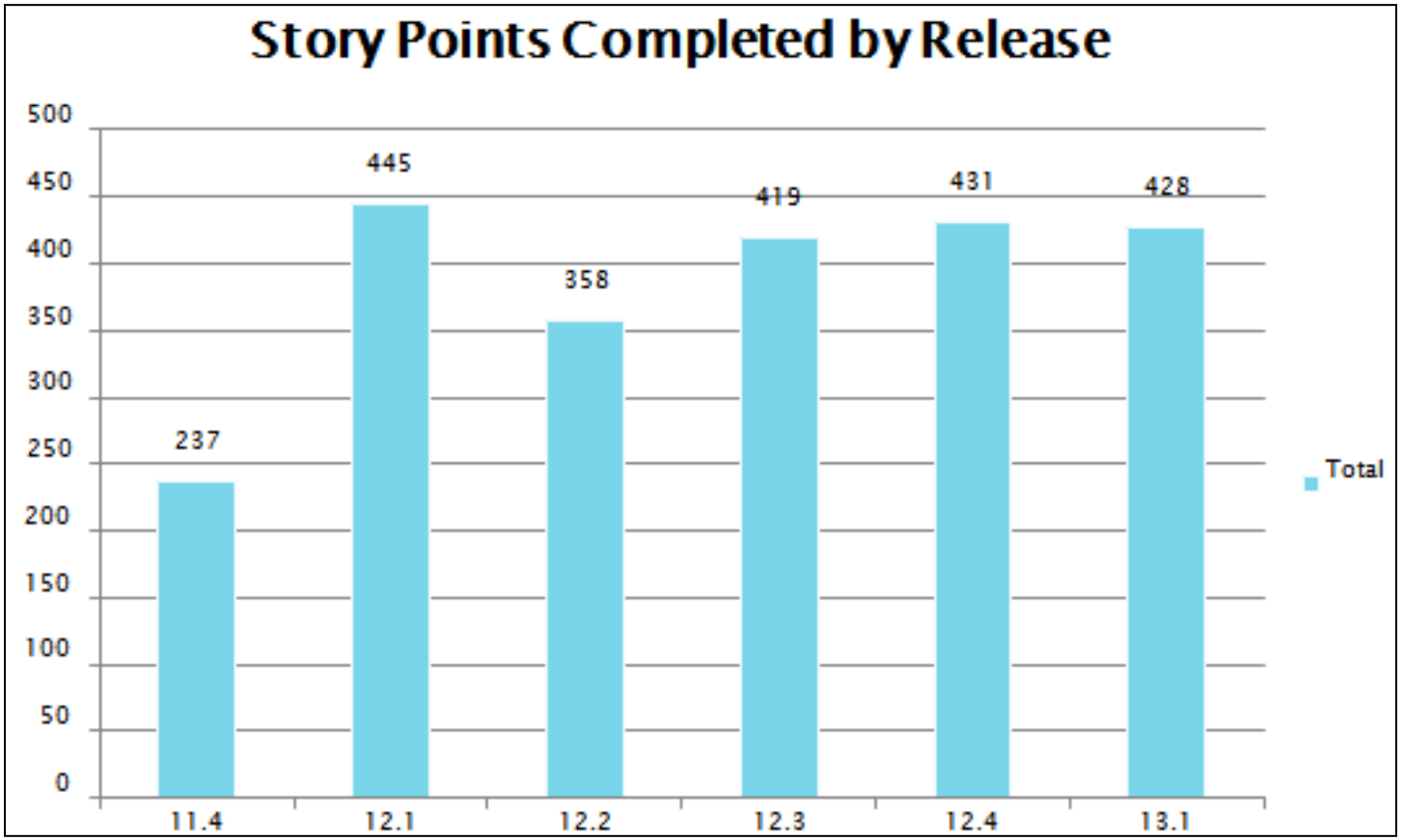

Process for Managing New Development
April
May
June
July
August
September
October
November
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
<2 weeks>
13 weeks
<2 weeks>
4 weeks
Sprint 1
Sprint 2
Sprint 3
Problem Reports from Planning Sessions
New PRs Added
<2 weeks>
Sprint Planning Start
Development Start
New Development Merge Points
Sprint Planning Start For 3rd Sprint
Pilot Candidate
Pilot
Statewide Release
Managers
Approve project
Functional Analysts
Specifiy requirements
Systems Analysts
Design and write specifcation (indefinite time)
Developers
Develop from specifications
Merge code
Quality Assurance
Test per design
Regression test
Enterprise Operations Team
DBA
Distribution Install
Distribution Install
Help Desk
User support
Trainers
Revise training curriculum, schedule & deliver training
<2 weeks>
13 weeks
<2 weeks>
4 weeks
Sprint 1
Sprint 2
Sprint 3
Problem Reports from Planning Sessions
New PRs Added
<2 weeks>
Sprint Planning Start
Development Start
New Development Merge Points
Sprint Planning Start For 3rd Sprint
Could shorten or eliminate this sprint based on the needs of new development.
Pilot Candidate
Pilot
Statewide Release
Release 11.2
Release 11.3
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
April
May
June
July
August
September
October
November


Potential Server Needs
January
February
March
April
May
June
July
August
September
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
– 2010 – Branch 5.6.1
<2 weeks>
4 weeks
Pilot Candidate
Pilot
Statewide Release
Release 11.2
<2 weeks>
13 weeks
<2 weeks>
4 weeks
Sprint 1
Sprint 2
Sprint 3
Problem Reports from Planning Sessions
New PRs Added
Sprint Planning Start
Development Start
Pilot Candidate
Pilot
Statewide Release
Release 11.3
<2 weeks>
13 weeks
<2 weeks>
4 weeks
Sprint 1
Sprint 2
Sprint 3
Problem Reports from Planning Sessions
New PRs Added
Sprint Planning Start
Development Start
New Development Merge Points
Pilot Candidate
Pilot
Statewide Release
Release 11.4
<2 weeks>
13 weeks
Sprint 1
Sprint 2
Problem Reports from Planning Sessions
Sprint Planning Start
Development Start
Server Needs
Historical statewide release - 5.5
QA 5.6a
Pilot - 5.6a
Historical statewide release - 5.6a
QA - 11.2
Pilot - 11.2.1
Historical statewide release - 11.2
QA - 11.3
Pilot - 11.3.1
QA - 11.4

Planned vs. Actual Release Dates

Release	Planned Date	Actual Date
11.4	November	November 2011
12.1	February	February 2012
12.2	May	May 2012
12.3	August	October 2012
12.4	November	December 2012
13.1	February	February 2013
13.2	May	May 28, 2013
13.3	August	

Story Points by Release



Developer Work Capacity by SSIS Release Cycle

Developer Specialty Area	Number of Developers	Percentage of Time Available: New Development	Percentage of Time Available: Problem Reports	Functional Area Allocation: Fiscal	Functional Area Allocation: Worker	Functional Area Allocation: Adoptions
GUI Developers	7	50%	50%	40%	40%	20%
Database Developers	4	70%	30%	40%	40%	20%
Architecture Developers	2	90%	10%	40%	40%	20%

		Weeks	Hours by Functional Area*: Fiscal	Worker	Adoptions
Release Cycle	Sprint 1	4	251	251	125
	Sprint 2	4	251	251	125
	Sprint 3	5	314	314	157
Total working hours by release cycle by functional area			816	816	407

	Story Points by Functional Area: Fiscal	Worker	Adoptions	Total
Story points by sprint by functional area	50	50	25	125
Story points by release cycle by functional area	163	163	81	407

*Formula for Hours by Functional Area = Number of developers * percentage of time available for PRs * functional area allocation * average working hours per week per developer * number of weeks in sprint

Average working hours per week per developer = 32

Number of hours in a story point = 5

Story Points by Release by Functional Area

Sum of Story Points	Area						
Release	1. Worker	2. Fiscal	3. Both	4. Adoptions	5. Architecture	6. None	Grand Total
12.1	214	176	30	3	20	2	445
12.2	165	120	49		24		358
12.3	274	99	31		15		419
12.4	208	130	37	24	32		431
13.1	269	94	41	27	24		455
Grand Total	**1130**	**619**	**188**	**54**	**115**	**2**	**2108**
Percent of Total	**53.6%**	**29.4%**	**8.9%**	**2.6%**	**5.5%**	**0.1%**	**100.0%**
						Average Story Points Per Release	**422**

Lessons Learned

- Establishing fixed release dates and hitting them builds credibility
- Fixed release dates facilitate work planning
- With credibility, users are more willing to negotiate schedule
- Project team functions more efficiently when a rhythm is established
- Software quality goes up as predictable release dates reduce pressure on the team

If we had to do it all over again, we would (fill in the blank)

- Establish a fixed release cycle as soon as possible
- Hit the published release dates – every time
  - Do not push last-minute fixes into the release
  - Negotiate scope at the front of each scheduled release
- Make it a priority to develop metrics measuring quantity and quality of the work

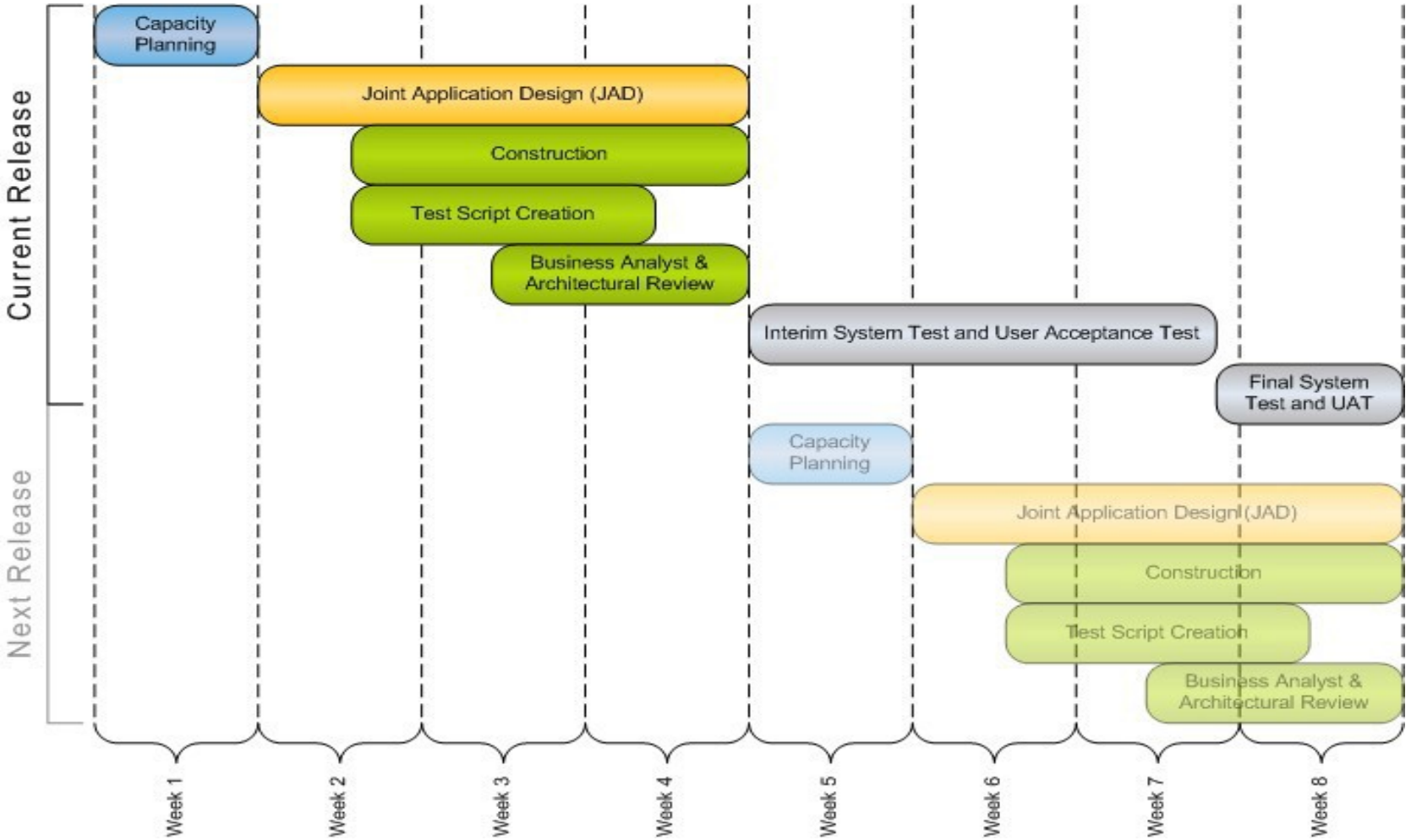
State Presentation: Ohio

- State background
- Development methodology
- Software framework
- System platform
- Impacts upon release management
- Lessons learned
- If we had to do it all over again, we would (fill in the blank)

State Background

- County administered (88 counties)
- Web-based
- Working on Federal compliance
- Mix of state and vendor development staff
- Integrated project team (Child Welfare and IT)
- Over 7,000 users
- New key initiatives
  - Private agencies Phase 2
  - IV-E court rollout
  - Field assistant mobile application
  - OptimalJ refactoring

Development Schedule



IPT Schedule

Week 1

Monday	Tuesday	Wednesday	Thursday	Friday
JAD Prep	JAD	JAD Prep / Design	JAD	Design
Construction	Construction	Construction	Construction	Construction
Create Test Scripts	Create Test Scripts	Create Test Scripts	Create Test Scripts	Create Test Scripts
Last Dev Day				
Dev / Test Kickoff	**Create Interim Build**	Deploy Interim to SYSTEST and UAT	Test SYSTEST & UAT	Test SYSTEST & UAT
			Commit Release	

Week 2

Monday	Tuesday	Wednesday	Thursday	Friday
JAD Prep / Design	JAD	JAD Prep / Arch Review	JAD / BA Review	Design
Construction	Construction	Construction	Construction	Construction
CGI SYSTEST	CGI SYSTEST	CGI SYSTEST	CGI SYSTEST / BA Review	CGI SYSTEST
Test SYSTEST & UAT	Test SYSTEST & UAT	Test SYSTEST & UAT	Test SYSTEST & UAT	Test SYSTEST & UAT
				Status Meeting

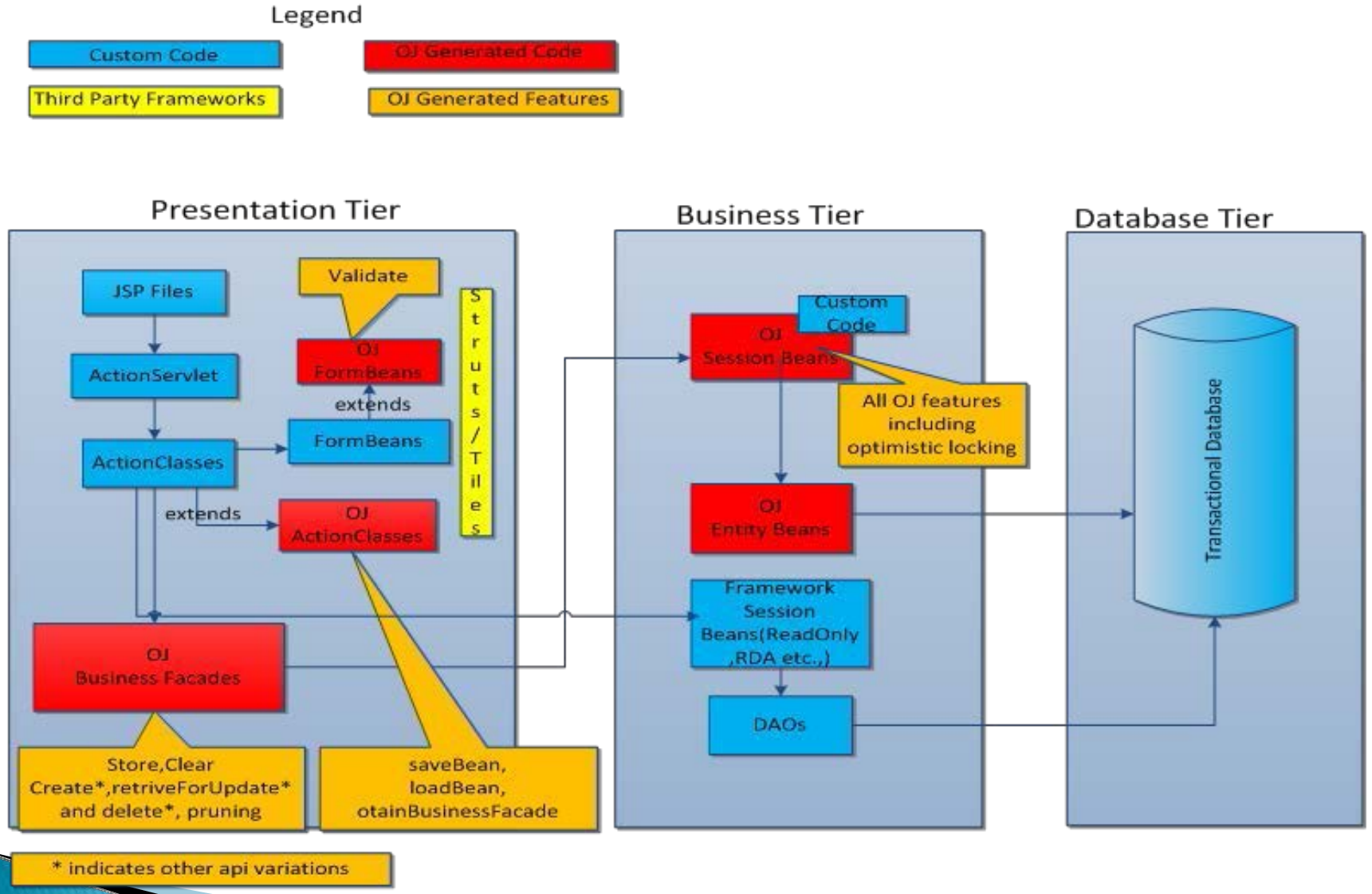
Week 3

Monday	Tuesday	Wednesday	Thursday	Friday
JAD Prep	JAD	JAD Prep / Design	JAD	Design
Construction	Construction	Construction	Construction	Construction
Create Test Cases	Create Test Cases	Create Test Data/Scripts	Create Test Data/Scripts	Create Test Data/Scripts
Last Fix Day	**Create Final Build**	Deploy to SYSTEST	Test SYSTEST	Deploy to UAT

Week 4

Monday	Tuesday	Wednesday	Thursday	Friday
JAD Prep / Design	JAD	JAD Prep / Arch Review	JAD / BA Review	Design
Construction	Construction	Construction	Construction	Construction
CGI SYSTEST	CGI SYSTEST	CGI SYSTEST	CGI SYSTEST / BA Review	CGI SYSTEST
Test UAT	Test UAT	Test UAT	**Deploy to Production**	**Deploy to Staging**

Software Framework & Platform



Impacts on Release Management

- Integrated team approach and Agile inspired practices offer flexibility in development process and schedule
- Upfront planning to incrementally introduce functionality ensures greater buy in and success for larger initiatives/complex functionality
- Release schedule milestones are checked regularly; methods for delaying releases and implementing hot fixes are in place in the event they are needed
- Release progress communicated to users in multiple venues: release note and database change documentation are provided to **users in advance**
- **Significant outreach to county users to participate in ongoing requirements documentation, design and testing process**

Lessons Learned

- Collaboration is “KEY”
- Users need mechanisms to provide constant feedback on functionality (user group meetings, testing build calls, help desk, survey data, focus groups, etc.)
- Self-managing teams meet daily to review progress and resolve issues
- A combination of manual and automated testing with collaboration of county, state and development staff
- Risk management identification, communication and mitigation on an ad hoc, daily and milestone basis
- If functionality is not being used or meeting needs, team stays open to “get it right”
- Team needs protocols for reducing scope or moving/adding resources to support critical priorities when issues arise

If we had to do it all over again, we would (fill in the blank)

- Streamline the front end to ease use for caseworkers in the field
- Plan for incremental roll out of modules (rather than entire system at once)
- Formalize coordination of dependencies between business functions and user groups during planning phase

IPT Process

Activity	Prepare for JAD
Goal	Finalize the pre-JAD Screen Specs
Participants/Role	➢IPT Leads
Tasks	➢Review Existing Materials ➢Conduct Preliminary Analysis ➢Develop Clarification Questions ➢Prioritize Screen Specs and/or Quality Center Items
Input(s)	➢Business Priorities ➢Quality Center Items ➢Preliminary Screen Specifications ➢Workflows
Output(s)	➢JAD Schedule ➢JAD Agenda ➢List of Questions

IPT Process

Activity	Conduct JAD Session/Requirements
Goal	To obtain consensus that each screen spec covered is clear, complete and ready for design
Participants/Role	➢IPT Leads ➢Business/Technical SMEs as required
Tasks	➢Conduct Walkthroughs of Screen Specifications and Report Mockups ➢Call for Consensus on Screen Spec Completeness ➢Update Screen Specs in Place (optional)
Input(s)	➢JAD Schedule ➢JAD Agenda ➢List of Questions ➢Preliminary Screen Specifications
Output(s)	➢Create / Update Quality Center items ➢Updated Screen Specifications ➢Use Case Consensus

IPT Process

Activity	Conduct JAD/BA Review
Goal	To obtain consensus that each screen specs design meets the intent of the requirements and is ready for construction
Participants/Role	➢IPT Lead ➢Business/Technical SMEs as required
Tasks	➢Conduct Walkthroughs of Screen Specs Realizations ➢Review Screen flows to confirm all possible paths and correct as required ➢Walkthrough Report flow and Content ➢Review Boundary Scenarios to Make Sure the Requirements and Design Handle Them Correctly
Input(s)	➢Updated Screen Specifications ➢Updated Report Specifications
Output(s)	➢Updated Quality Center items ➢Updated Screen Specifications ➢Updated Report Specifications ➢Optimal/J Artifacts

IPT Process

Activity	Conduct Design
Goal	To review requirement documents and develop a technical design that meets the intent of the requirements
Participants/Role	➢IPT Leads ➢Business/Technical SMEs as required
Tasks	➢Review Screen Specs / Quality Center Items ➢Compare Screen Flows to Requirements ➢Review Screen and Report Specifications for Each Screen and Report Related to the Screen Spec ➢Review Boundary Scenarios ➢Create Optimal/J Artifacts
Input(s)	➢Update Quality Center Items ➢Updated Screen Specifications ➢Use Case Consensus
Output(s)	➢Optimal/J Artifacts

IPT Process

Activity	Conduct Architecture / Design Round Table
Goal	To obtain consensus that technical design meets the SACWIS architecture guidelines and module inter-dependencies are resolved
Participants/Role	➢IPT Leads ➢Additional SACWIS Team Leads
Tasks	➢Share and Review Technical Design ➢Discuss Inter-dependencies Across Modules
Input(s)	➢Optimal/J Artifacts ➢Updated Quality Center items ➢Updated Screen Specifications
Output(s)	➢Design Consensus ➢Updates to Optimal/J Artifacts ➢New Requirements / Updates to Frameworks

IPT Process

Activity	Construction
Goal	To develop detailed design and code artifacts that meets the intent of the requirements, high level design and is ready for UAT
Participants/Role	➢IPT Development Team ➢IPT Leads
Tasks	➢Review Requirements and High Level Design Documents ➢Develop Detailed Design / Implementation Approach ➢Review Implementation Approach with IPT Leads ➢Develop Code Artifacts and Complete Unit Testing ➢Review Code Artifacts with IPT Leads as Required
Input(s)	➢Updated Optimal/J Artifacts ➢Updated Quality Center Items ➢Updated Screen Specifications ➢Update Report Specifications
Output(s)	➢Code Artifacts ➢Update Quality Center Items

IPT Process

Activity	Internal System Testing
Goal	To develop test scripts and data population scripts that verifies whether the code artifacts meets the intent of the requirements
Participants/Role	➢IPT Test Team ➢IPT Business Lead
Tasks	➢Review Requirements and High Level Design Documents ➢Develop Test Scripts ➢Develop Data Population Scripts ➢Review Test Scripts with IPT Business Lead
Input(s)	➢Quality Center items ➢Updated Quality Center items ➢Updated Screen Specifications ➢Update Report Specifications
Output(s)	➢Test Scripts (uploaded to Quality Center) ➢Data Population Scripts

Supporting Information

- Classics
  - http://www.projectsmart.co.uk/docs/chaos-report.pdf
    - A copy of the classic 1995 report on SW project failure by the Standish Group
- Agile
  - http://www.agilealliance.org
    - The home of the Agile Alliance, with a great library of Agile articles
  - http://www.mountaingoatssoftware.com/system/presentationfile/52/SDWest2007_EUS.pdf
    - Effective user stories by Mike Cohen
  - http://www.bluecollarobjects.com/pub/Main/Agile2009/Federal_Bureaucracy-4.pdf
    - A Retrospective: Managing Agile Transition in Government Bureaucracy by Brandon Raines and Judy Wankerl

Supporting Information

- Agile (cont.)
  - http://www.poppendieck.com
    - Mary and Tom Poppendieck's home website, excellent material on Lean
  - http://xpday3.xpday.org/slides/LeanTutorial.pdf
    - Overview and Tutorial on Lean by Mary Poppendieck
  - http://leansoftwareengineering.com
    - Good articles on Lean from multiple authors
  - http://leansoftwareengineering.com/ksse/scrum-ban/
    - On Kanban workflow by Corey Ladas, author of Scrumban

Attendee Discussion

Wrap Up

- What was accomplished today?

- What's next?

- Reminder: Recorded versions of each of the six webinars are being made available at: http://www.acf.hhs.gov/programs/cb/research-data-technology/state-tribal-info-systems/training