



Use of Agile System Validation Methodologies for Rapid Product Enhancement

Terek Peterson
Giri Balasubramanian



Transforming Clinical Trials

- **What is Agile at PRA?**
- **EXACT Product Lifecycle Management using Agile**
- **Agile Key Definitions**
- **Three concurrent activities in each development cycle**
- **Application of Agile in EXACT development**
- **Comparison with prior SDLC**
- **Example of a Sprint enhancing usability**
- **Internal Quality Audit**
- **Benefits to our sponsors**
- **Conclusions/Questions**



System Development Today

- ✓ *Business today is fast paced and dynamically changing*
- ✓ *Over the years the software industry has come to learn it's almost impossible to predict the future*
- ✓ *Most people are unable to visualize what the application will look like or work from specifications*
- ✓ *Complexity of development will always surface unexpected risks*
- ✓ *To be successful in developing valuable, useful software, you must be Agile and use a discovery based process*
- ✓ *That incrementally provides the “customer” with something they can visualize and provide feedback along the way*
- ✓ *Avoids big-bang releases but delivers periodic, incremental value to the customer*



What is Agile at PRA?

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Agile at PRA helps achieve this through:

- ✓ *Disciplined Project Management Process*
- ✓ *Frequent inspection and adaptation*
- ✓ *Leadership philosophy that encourages teamwork*
- ✓ *Self-organization and accountability*
- ✓ *Rapid delivery of high-quality software*
- ✓ *Alignment of development with customer needs and company goals*



Agile Key Definitions

Item	Definition
Burndown graph	The trend of work remaining across time in a Sprint, a release, or a product. The source of the raw data is the Sprint Backlog and the Product Backlog, with work remaining tracked on the vertical axis and the time periods (days of a Sprint or Sprints) tracked on the horizontal axis.
Chicken	Someone who is interested in the project but does not have formal Scrum responsibilities and accountabilities (is not a Team member, Product Owner, ScrumMaster, or other stakeholder).
Daily Scrum meeting	A short status meeting held daily by each Team during which the Team members synchronize their work and progress and report any impediments to the ScrumMaster for removal.
Done	Complete as mutually agreed to by all parties and conforming to an organization's standards, conventions, and guidelines. When something is reported as "done" at the Daily Scrum or demonstrated as "done" at the Sprint review meeting, it must conform to this agreed definition.
Estimated work remaining	The number of hours that a Team member estimates remain to be worked on any task. This estimate is updated at the end of every day the Sprint Backlog task is worked on. The estimate is the total estimated hours remaining, regardless of the number of people that perform the work.
Increment	Product functionality that is developed by the Team during each Sprint.
Increment of potentially shippable product functionality	A completely developed increment that contains all of the parts of a completed product, except for the Product Backlog items that the Team selected for this Sprint.



Agile Key Definitions (*cont.*)

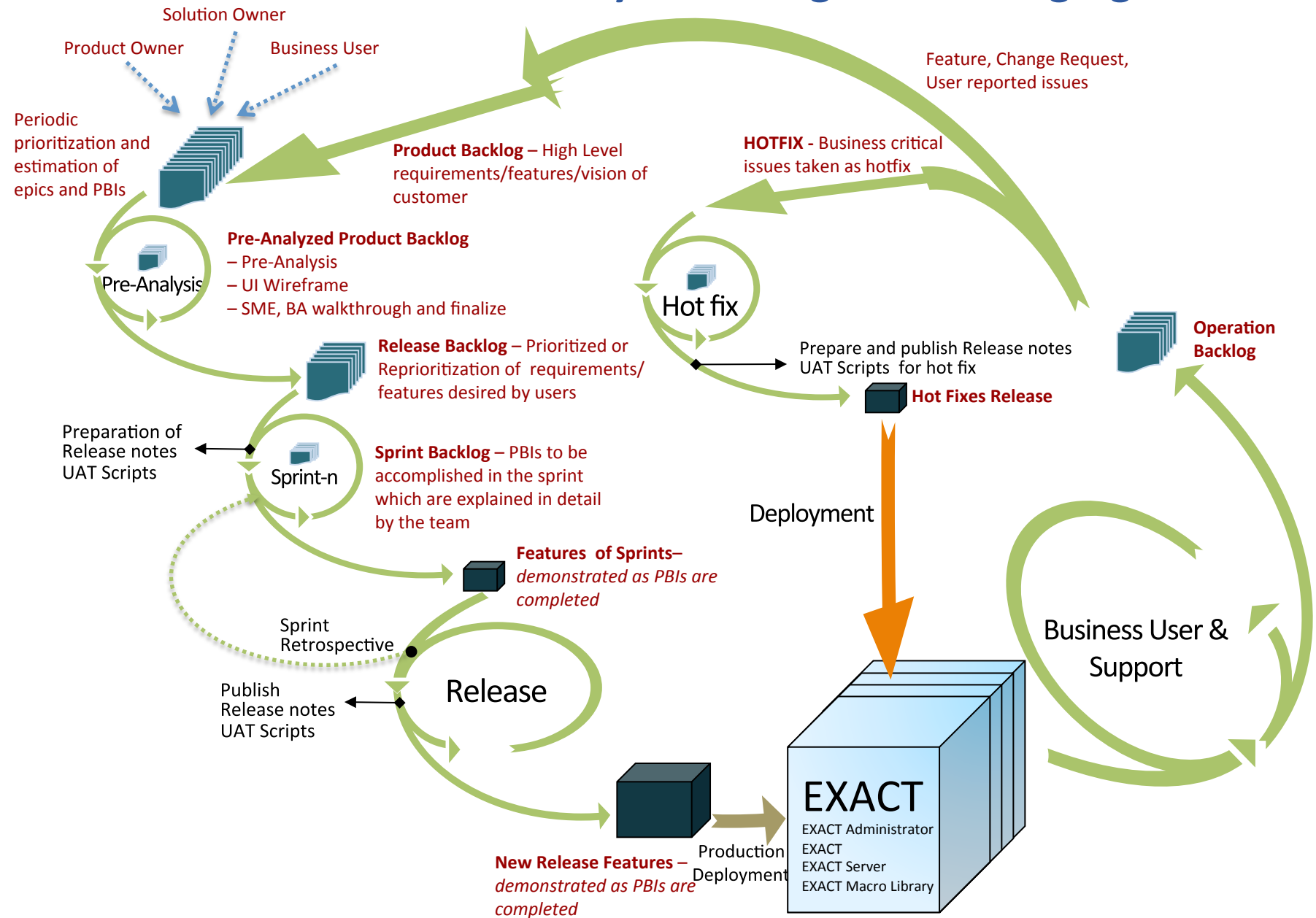
Item	Definition
Pig	Someone occupying one of the three Scrum roles (Team, Product Owner, ScrumMaster) who has made a commitment and has the authority to fulfill it.
Product Backlog	A prioritized list of project requirements with estimated times to turn them into completed product functionality. Estimates are in days and are more precise the higher an item is in the Product Backlog priority. The list evolves, changing as business conditions or technology changes.
Product Backlog items	Functional requirements, nonfunctional requirements, and issues, which are prioritized in order of importance to the business and dependencies and then estimated. The precision of the estimate depends on the priority and granularity of the Product Backlog item, with the highest priority items that can be selected in the next Sprint being very granular and precise.
Product Owner	The person responsible for managing the Product Backlog so as to maximize the value of the project. The Product Owner represents all stakeholders in the project.
Scrum	Not an acronym, but mechanisms in the game of rugby for getting an out-of-play ball back into play.
ScrumMaster	The person responsible for the Scrum process, its correct implementation, and the maximization of its benefits.
Sprint	A time-box of 30 sequential calendar days during which a Team works to turn the Product Backlog it has selected into an increment of potentially shippable product functionality.
Sprint Backlog	A list of tasks that defines a Team's work for a Sprint. The list emerges during the Sprint. Each task identifies those responsible for doing the work and the estimated amount of work remaining on the task on any given day during the Sprint.



Agile Key Definitions (*cont.*)

Item	Definition
Sprint Backlog task	One of the tasks that the Team or a Team member defines as required to turn committed Product Backlog items into system functionality.
Sprint planning meeting	A one-day meeting time-boxed to 8 hours that initiates every Sprint. The meeting is divided into two 4-hour segments, each also time-boxed. During the first segment, the Product Owner presents the highest priority Product Backlog to the Team. The Team and the Product Owner collaborate to help the Team determine how much Product Backlog it can turn into functionality during the upcoming Sprint. The Team commits to this Product Backlog at the end of the first segment. During the second segment of the meeting, the Team plans how it will meet this commitment by detailing its work as a plan in the Sprint Backlog.
Sprint retrospective meeting	A meeting time-boxed to 3 hours and facilitated by the ScrumMaster at which the Team discusses the just-concluded Sprint and determines what could be changed that might make the next Sprint more enjoyable or productive.
Sprint review meeting	A meeting time-boxed to 4 hours at the end of every Sprint at which the Team demonstrates to the Product Owner and any other interested parties what it was able to accomplish during the Sprint. Only completed product functionality can be demonstrated.
Stakeholder	Someone with an interest in the outcome of a project, either because he or she has funded it, will use it, or will be affected by it.
Team	A cross-functional group of people that is responsible for managing itself to develop software every Sprint.
Time-box	A period of time that cannot be exceeded and within which an event or meeting occurs. For example, a Daily Scrum meeting is time-boxed to 15 minutes and terminates at the end of those 15 minutes, regardless.

EXACT Product Lifecycle Management using Agile



Sprints are the basic unit of development during which a Team works to turn the Product Backlog (requirements) into an increment of potentially shippable product functionality within fixed duration



Three Concurrent Activities in Each Development Cycle

Three concurrent activities in lifecycle:

✓ **Product Backlog** – Develop requirements

- Solution Leader/owner creates requirements and user stories/product backlog from user requests and needs
- Contains high level requirements, user stories/product backlog items and defects that apply to the product.
- Groomed and prioritized frequently to keep all items pertinent.

✓ **Pre-Analyzed Product Backlog** – Review & walk through the requirements

- PBIS reviewed as a team, questions answered, PBIS refined by SMES and Solution Leader/Owner
- UI Designs reviewed and refined by Technical Architect / Business Analyst
- Prioritization re-evaluated based on effort
- Sprint Backlog created based on team velocity
- Design created with Architectural Guidance / Database Schema designed
- Solution Design Finalized by solution Leader/ owner

✓ **Sprint Backlog** – Develop and Testing (QC)

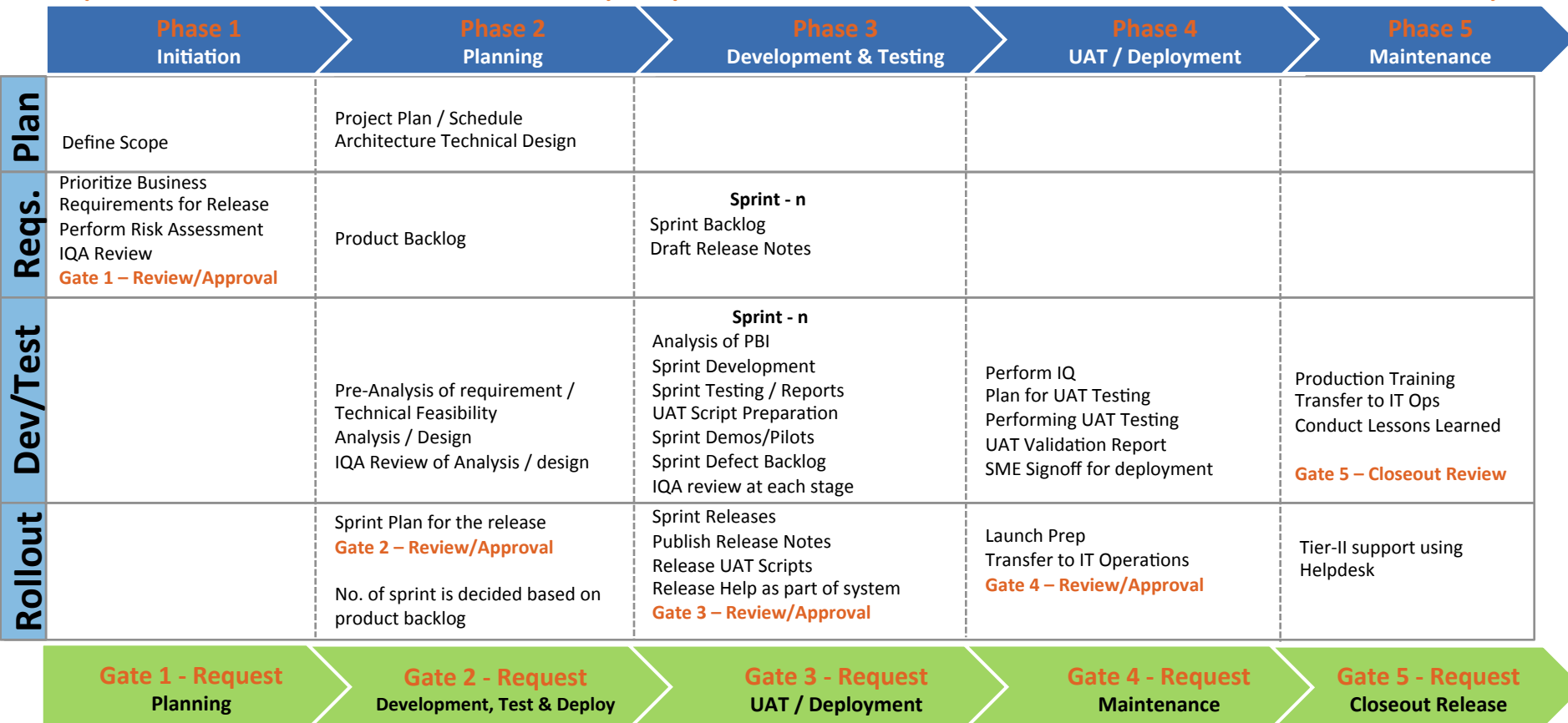
- Analysis of the prioritized product backlogs
- Build/Code features, Code Review, Unit Test
- Functional/System Test Cases Executed – Defects created, triaged and fixed
- Documentation/Help Created and Validated
- Features are tested with features across sprints, Defects created, triaged and fixed
- Solution Owner decides whether sprint features are to be demoed, user tested or UAT
- Deployment Plans created, Features deployed to environments accordingly
- Sprint Review with SMEs & Solution Lead/ owner and Demo feedback and defects are captured for future sprints or releases



EXACT Detail Development Lifecycle

Release Planning

Sprint Execution (Monitor & Control)



Backlog – Requirements captured from solution owner and business users

Gate Approval – to ensure that project is ready to move forward and that all pertinent topics have been considered, reviewed, and addressed as appropriate.

IQA – Internal Quality Audit

UAT – User Acceptance Test

Tier – II Support – Support provided by Kinship team. HD Tickets pertaining to issues , unable to analyze and conclude whether it is a clarification will fall under Tier II support team



EXACT Lifecycle - Detail Sprint Activity (Cont.)

Sprint Preparation

Planning

- Release Backlog Created/Updated
- PBI Created in PBI Backlog – Supporting Documents, Wireframes created and attached to PBIs
- Release Backlog Prioritized according to Business Value

Architecture Review

- Review Requirements for Architectural Approach and Enterprise Governance
- PBI return to planning if any push back

Pre-Analysis

- Flush out PBI details
- High Level PBI Points added to requirement in Release Backlog
- Data Model is perceived
- Solution Lead signs off on PBI and wireframes

Sprint Execution

Analysis

- PBI reviewed as a team, questions answered, PBI refined
- UI Designs reviewed and refined
- Points reevaluated
- Sprint Backlog created based on team velocity
- Draft Release Notes, RTM prepared

Design

- Design created with Architectural Guidance
- Database Schema designed
- Solution Design Finalized
- Functional/System Test, Integration Test Case Created
- UAT Scope prepared, RTM Prepared

Development and Test

- Help Created, UAT Script Prepared
- Unit Test Case Design, Build/Code features, Code Review, Unit Test
- Functional/System, Integration Test Cases Executed – Defects created and fixed
- Documentation and Validated
- UAT script prepared, RTM executed

Regression Testing/UAT Dry Run

- Features are tested with features across PBIs and sprints
- Defects created and fixed
- UAT scripts are dry run to ensure the PBI implementation, defects created and fixed

Sprint Close

Sprint Review and Demo

- Sprint reviewed with Solution Lead
- Feedback is resolved or moved to future PBI or defects

Sprint Retrospective

- Discuss what worked well, what didn't
- Determine changes to be made
- Scrum master takes notes and manages changes

Deployment/ Smoke Testing / User Testing

- Solution Owner decides whether sprint features are to be demoed, user tested or UAT
- Deployment Plans created
- Features deployed to environments accordingly
- Smoke Testing is performed in the deployed
- Feedback and defects are captured for future sprints or releases



Application of Agile in EXACT Development

to deliver cost-savings and automation to sponsors

- ✓ **Quick way of meeting business needs with quality by way of releases over a period**
- ✓ **Process driven development being followed**
- ✓ **Methodology facilitates prioritization of business requirement periodically**
- ✓ **Methodology facilitates release of new features**
- ✓ **Adaptability to constant business requirement changes**

Comparison with Prior SDLC

DSDM

- ✓ Dynamic systems development method (DSDM) is a RAD approach to software development
- ✓ Iterative and Incremental
- ✓ Eliminates problems of:
 - ✓ Going over-budget
 - ✓ Missing deadlines
 - ✓ Users not involved
 - ✓ Management not committed
 - ✓ Work on key Principles like active user involvement, teams must take own decision, high level requirements should be unchanged etc.
- ✓ DSDM only demands strict obedience to the 9 principles
- ✓ DSDM does not aim to solve all development problems, when the 9 principles can not be implemented, DSDM is most likely not the best choice

Agile

- ✓ Agile methodology is an alternative to waterfall, or traditional sequential development.
- ✓ Short incremental, iterative work cadences, known as sprints.
- ✓ Agile offers 3 main frameworks which are Scrum, Lean Kanban and eXtreme Programming (XP)
- ✓ Framework is selected based on the industry need
- ✓ Agile follows some key principles like satisfy the customer through early and continuous delivery, welcome changing requirements, work together daily, face-to-face conversation etc.
- ✓ Agile Methodology is time saving and cost effective due to clear requirement and early detection of issues by regular feedback from the end user
- ✓ Transparency to each individual working on the project with the status of each phase

- ✓ **DSDM** was developed in the United Kingdom in the 1990s by a consortium of vendors and experts in the field of Information System (IS) development
- ✓ In 1974, a paper by E. A. Edmonds introduced an adaptive software development process and deployed at New York Telephone Company's Systems Development Center.
- ✓ **Agile** software development methods was developed in 2001 by a group of software experts who got together in the Snowbird resort of Utah to draft "Agile Manifesto"



Some Successes / Some Challenges

Some Successes

- ✓ Requirement to be implemented is very clear
- ✓ Successful short-term release
- ✓ Meet timeline
- ✓ Gather user feedback while in the sprint and implement during the sprint

Some Challenges

- ✓ Not able to run parallel sprint as code is tightly coupled
- ✓ Team members availability and openness during the scrum meeting
- ✓ Business users time availability for analysis, demo and feedback
- ✓ Change in priority by business which leads to Inclusion of PBIs after start of sprint or dropping of PBI due to technical constraint/challenge during sprint
- ✓ High number of bugs during development and testing
- ✓ Qualification of resource included in the sprint and Reallocation/ reshuffling of resource during the sprint
- ✓ More time taken for Scrum meeting as the team size is bigger (Average 15)
- ✓ Test automation of the product
- ✓ Maintenance of documentation when the release is short

Example of a Sprint Enhancing Usability

Mapping Interface in EXACT

- Facilitates the user to map clinical trial data to submission data model (SDTM/ADaM). It also enables the user to derive or transform the data in this process.

Features of Mapping in EXACT

- There are different option to map clinical trial data to submission data model
 - Auto Mapping – Auto maps identical trial data with submission data
 - Manual Mapping – Allows user to manually map trial data with submission data
 - Derivation Mapping – Allows user to map the data using derivation
 - Auto Sequence Number Generation – Allows user to automatically generate the sequence number
- Three types of mapping
 - One to One Mapping
 - One to Many Mapping
 - Many to One Mapping (for SUPPQUAL and CO domains)

Example of a Sprint Enhancing Usability (Cont..)

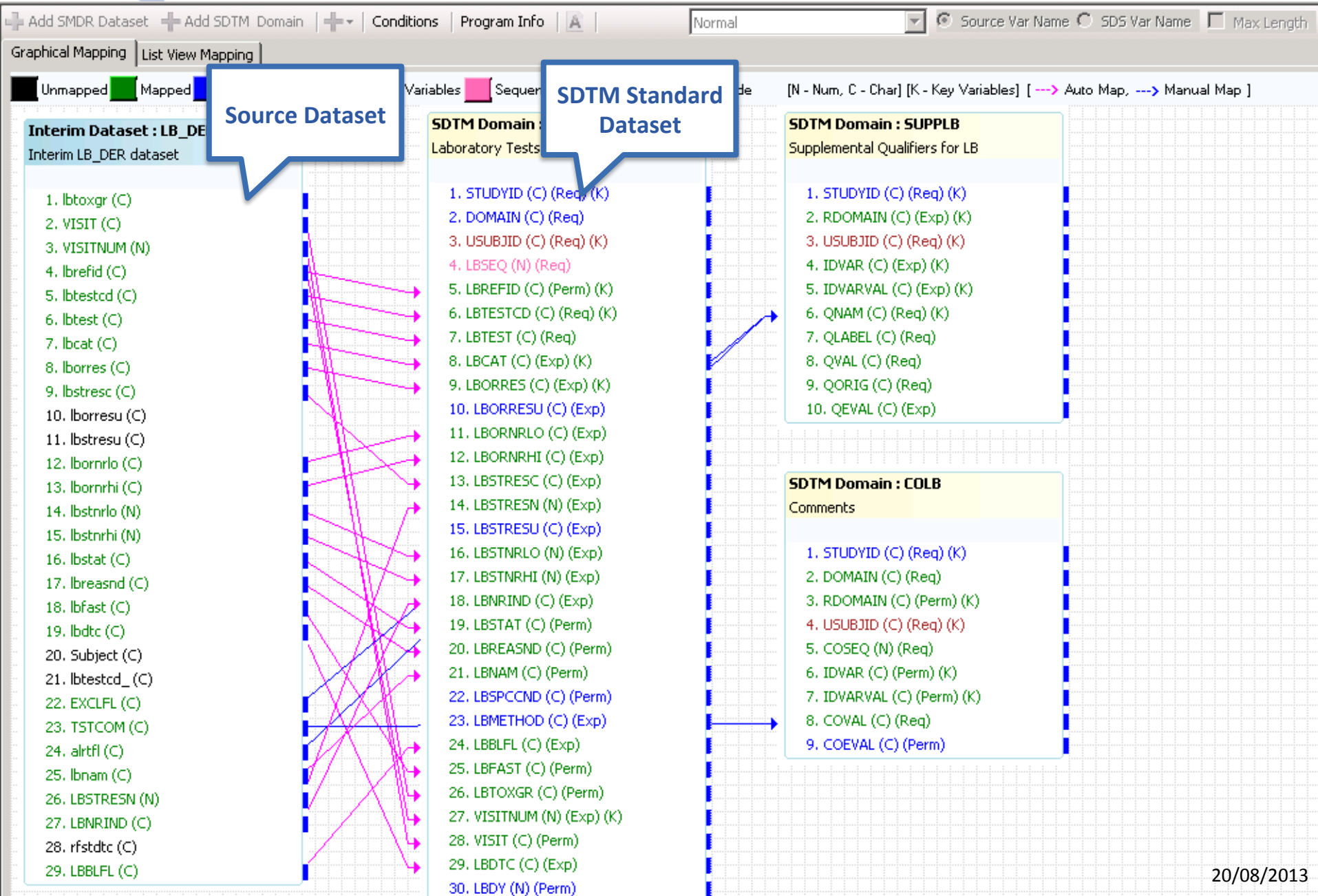
Feedback from user *before* redesign:

- ✓ User had to alternate between left and right clicking when looking to edit a variable regardless of how it is populated.
- ✓ Drag and drop option to map variables in interface is very tedious when there are more than 30 variables since the list below is not visible to.
- ✓ When several variables are available like for ADAE domain, it was not easy to see all the variables listed on the destination side because the bottom window will keep popping up for each variable
- ✓ User have to right click on the variable to add/edit mapping details after matching the variable by drag and drop
- ✓ Mapping lines are partially shown when domain and dataset are Straightly Mapped
- ✓ User would click somewhere on the screen and the bottom pop up will keep creeping up and user have to keep clicking the pushpin to hide that.

Benefits *after* redesign:

- ✓ Provides EXCEL like grid feature to add/edit a mapping of a domain variable with in same window
- ✓ View/Edit all mapped/unmapped variables and its mapping details configurations in a domain on same screen with a single click save/update
- ✓ View/Edit SUPPXX and COXX domains of a SDTM domain on same screen
- ✓ Avoids too many user click activity and avoids unnecessary pop ups

Example of a Sprint Enhancing Usability (Cont.)



Example of a Sprint enhancing Usability (Cont..)

+ Add SMDR Dataset + Add SDTM Domain + Conditions Program Info A Normal Source Var Name SDS Var Name Max Length

Graphical Mapping List View Mapping

Search Search Columns Remove Sorting LB

Domain	SDTM Variable Name	SDTM Variable Label	Length (SDTM)	SMDR Variable Name	SMDR Variable Label	Format	Input Format	Input F...	Output...	Output...	E...
L...											
STUDYID (C) (Req) (K)	Study Identifier	12									
DOMAIN (C) (Req)	Domain Abbreviation	2									
USUBJID (C) (Req) (K)	Unique Subject Id...	24									
LBSEQ (N) (Req)	Sequence Number	8									
LBREFID (C) (Perm) (K)	Specimen ID	75		lbrefid (C)	TCAC...						

Search Search Columns Remove Sorting

CO Domain	CO Variable Name	CO Variable Label	Identif...	Length (CO)	Source Variable
COLB					
STUDYID (C) (Req)...	Study Identifier	12			
DOMAIN (C) (Req)	Domain Abbreviation	2			
RDOMAIN (C) (Per...	Related Domain Abb...	2			
USUBJID (C) (Req) ...	Unique Subject Id...	24			
COSEQ (N) (Req)	Sequence Number	8			

Search Search Columns Remove Sorting SUPPLB

SUPP Domain	SUPP Variable Name	SUPP Variable Label	Length (SUPP)	Source Variable Name	Source Variable Label	Qualifier Variabl...	Qualifier Variabl...	QLABEL
SUPPLB								
STUDYID (C) (Req)...	Study Identifier	12						
RDOMAIN (C) (Exp ...	Related Domain Abb...	2						
USUBJID (C) (Req) ...	Unique Subject Id...	24						
IDVAR (C) (Exp) (K)	Identifying Variable	8						
IDVARVAL (C) (Exp ...	Identifying Variable ...	8						

Derivation Code DM.STUDYID Main Domain CO Domain SUPP Domain

LB. STUDYID (C) STUDYID = 'RDEA594-303';

LB. DOMAIN (C) DOMAIN = 'LB';

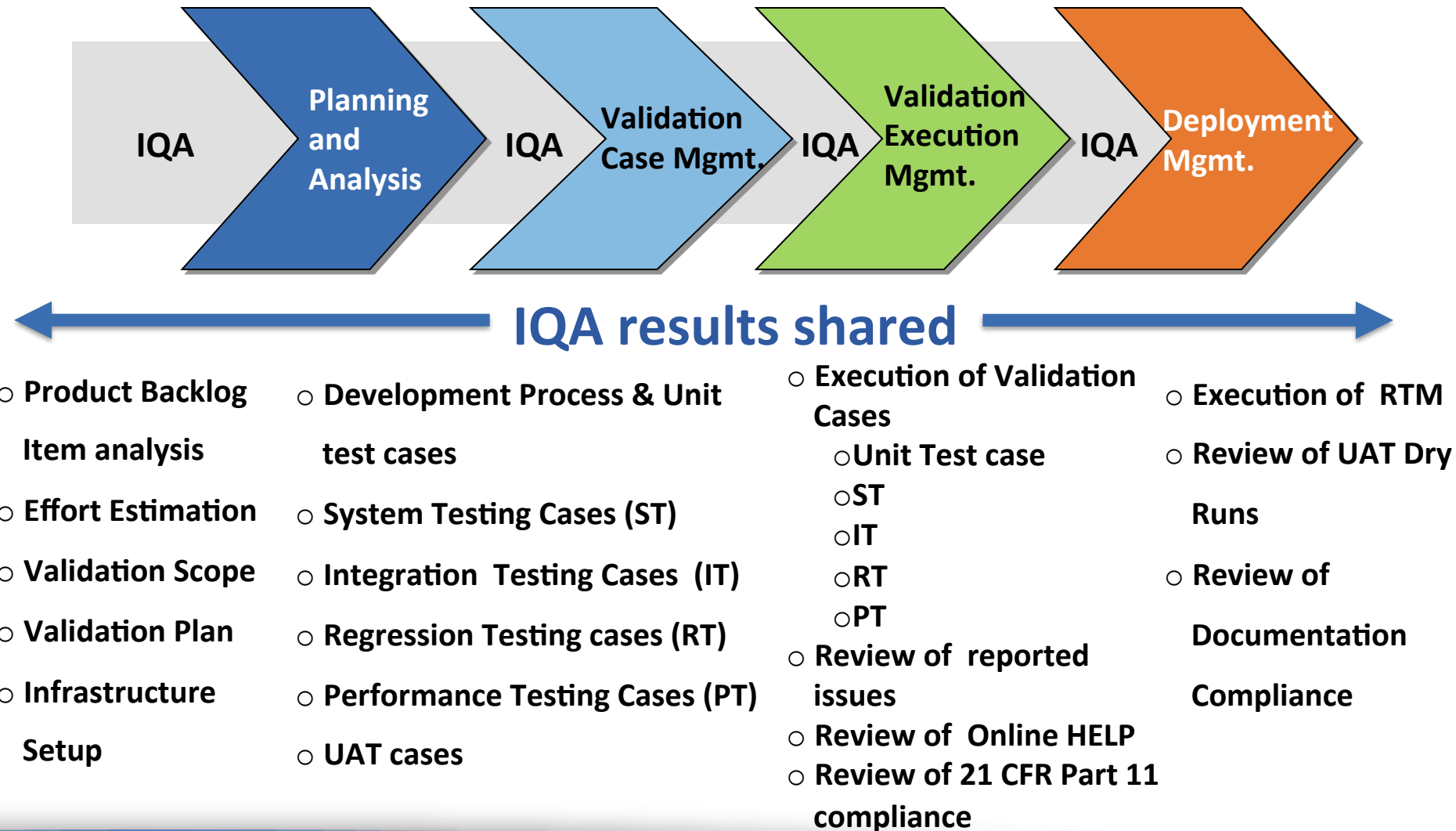
LB. USUBJID (C)

LB. LBSEQ (N)

- Feature to Sort variable based on the column
- Search for data in the column
- Freeze pane and work as in excel
- Re-order columns by dragging and dropping
- Enter code in the same interface
- Perform mapping main domain, SUPQUAL Domain and CO domain in single shot



How is Quality Management Organized as it Relates to Software Quality?





Benefits to Our Sponsors

Acceptance of EXACT increased with each release in short term in areas like

- *Concurrently work on more studies within the same timeframe*
- *Process driven framework*
- *Creation of Clinical Data Repository*
- *Built in SDTM compliance and validated creation of Define.xml*



Conclusions/Questions

Challenges

- ✓ *Quality of resources working in sprint should be available to achieve short-term release*
- ✓ *Resource back-up to be available*
- ✓ *Resource should have complete knowledge of product*
- ✓ *Continuous training and release of training documents*
- ✓ *Monitoring the quality of resource*
- ✓ *Continuous update to Regulatory Compliance (CDISC releases)*
- ✓ *User expectation with respect to Responsive Software Design*

- **Methodology is suitable for Regulatory compliant product development lifecycle**
- **Allows quick release of customized features to sponsors/business use and also complies with regulatory requirements**

The background of the slide is a dense collection of butterflies. Most are yellow with dark brown or black borders on their wings. One butterfly, positioned in the upper-middle left, is a vibrant blue with black markings on its wings. In the top right corner, there is a light blue rounded rectangle with a dark blue border.

**Thank
You!**



A Clear Difference