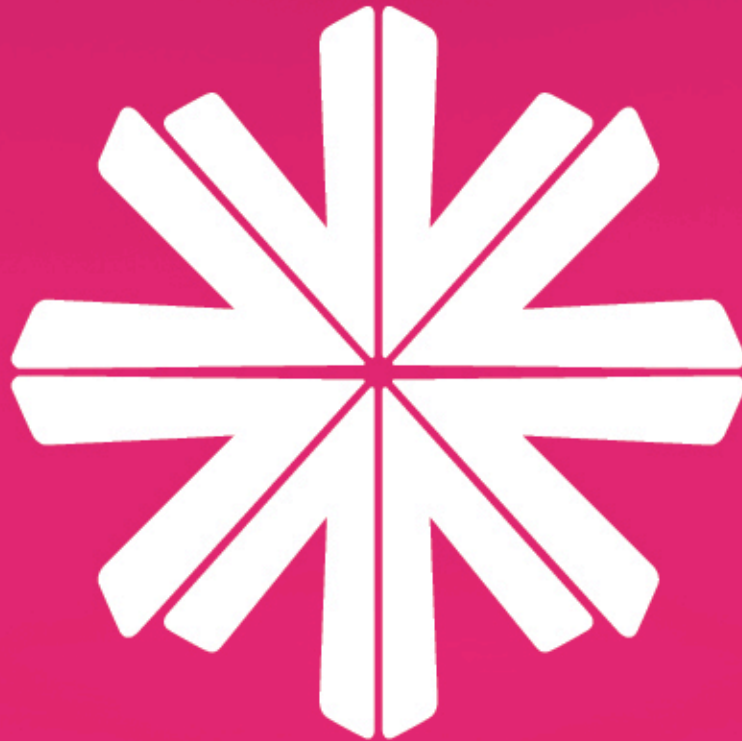


THE FUTURE OF
CLINICAL DEVELOPMENT



PRAHEALTHSCIENCES

Barcelona 2016

**Fast Track to Approval:
Speed and Efficiency**
9th–12th October

Princesa Sofia Gran Hotel, Barcelona



PRAHEALTHSCIENCES

Paper AD11 – PhUSE 2016

Clinical SAS Application Development in a GRID Environment

Giri Balasubramanian, PRA Health Sciences, Chennai, India

Edwin Ponraj Thangarajan, PRA Health Sciences, Chennai, India



- Session Scope
- Overview of SDLC Models
- Case Study I
- Case Study II
- Benefits

Session Scope

Paper Abstract



Application Development Methodologies and Process has been evolving over a period of time with the changing need & demand from business community. Transformation of SDLC Methodologies has started from Waterfall model in 1980's to Agile in 1990's and then to Lean model and DevOps Implementation with Agile Principles. Each of the models has its own merits & demerits and is judged based on agility that is needed for a kind of business. The session would be dwelling these models that are prevalent and bring about a case study to demonstrate the agility of recent models for development of SAS® based application development using different programming languages such as C# which needs to be deployed in a SAS® GRID Environment. Case study would focus on how a most recent methodology say DevOps is adopted to ever changing needs of users in building a SAS® based application and what kinds of toolsets, processes to be in place while still adhering to Agile principles for continuous integration, delivery & deployment of solutions



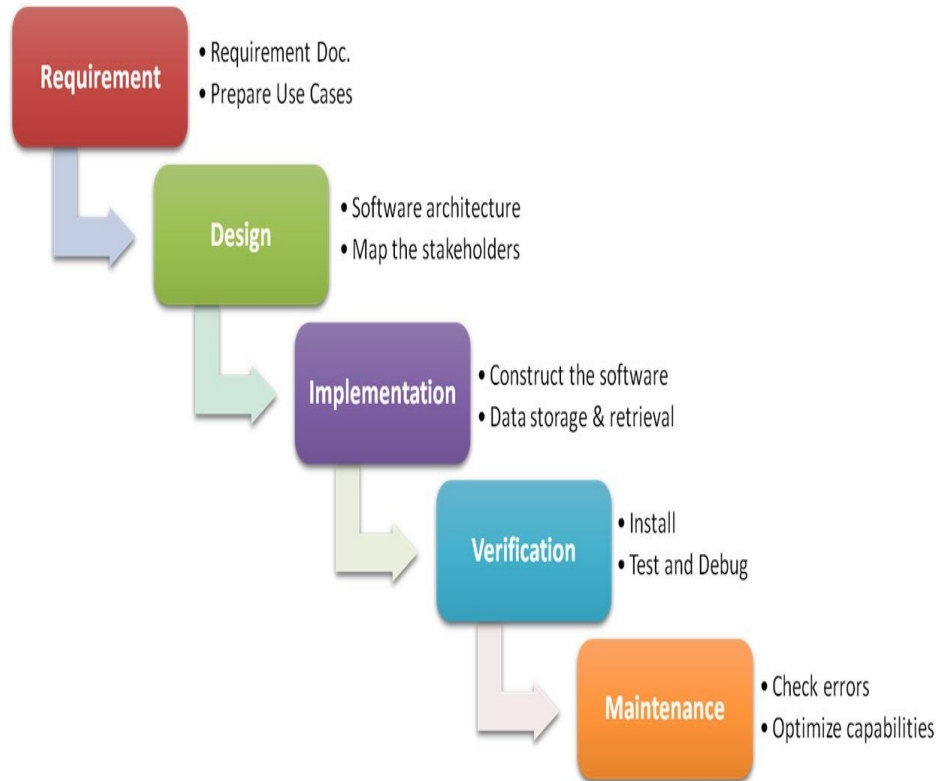
Application Development Methodology & Process

How to apply for Clinical SAS Application
Development in a GRID Environment?

- A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.
- The intent of a SDLC process is to help produce a product that is cost-efficient, effective, and of high quality.
- The SDLC methodology usually contains the following stages:
 - Analysis (requirements and design),
 - Construction,
 - Testing,
 - Release,
 - and maintenance (response)

Overview of SDLC Models

Traditional - Waterfall

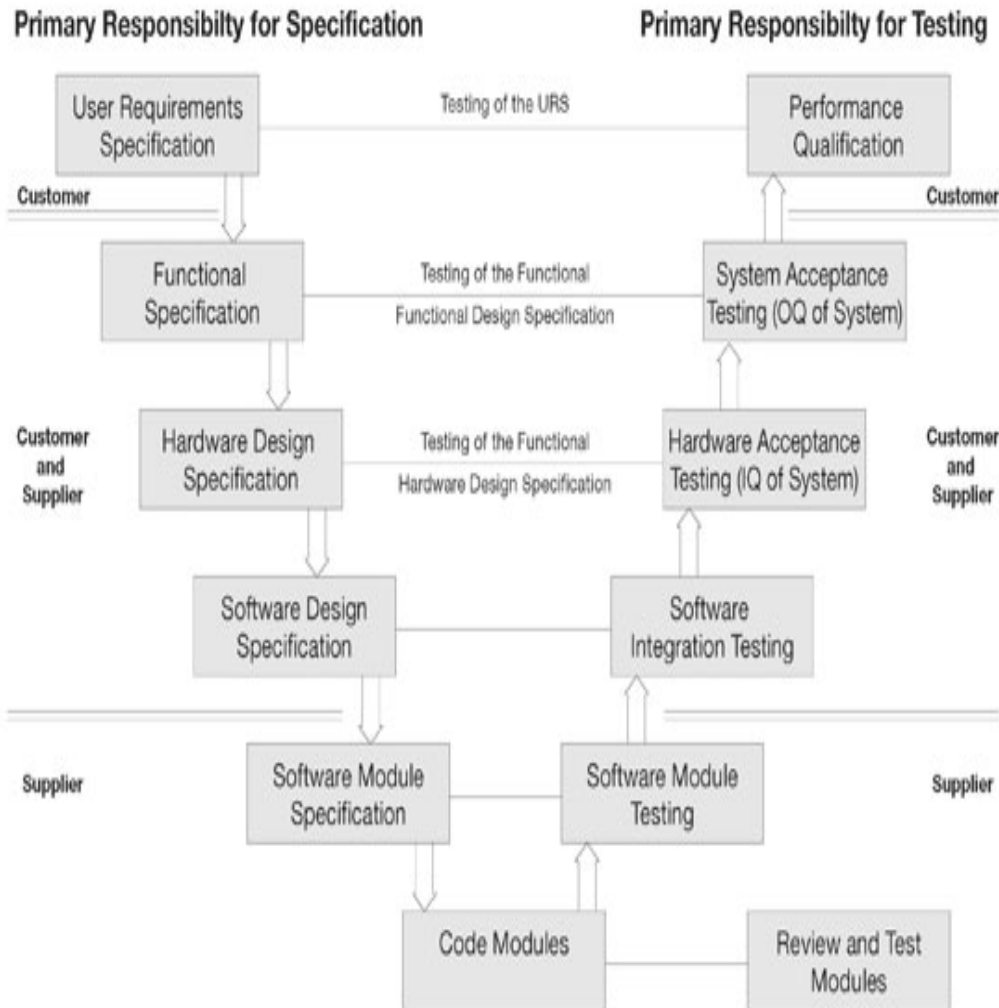


Waterfall is a linear - sequential development approach, in which development is seen as flowing sequentially through phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance. The sequential phases in waterfall model are Requirement Gathering & Analysis, System Design, Implementation, Integration & Testing, Deployment of System, and Maintenance. Each of these phases would start with an input from the earlier phase. Waterfall model was tweaked by the users into different forms to include iterative model into the sequential process.

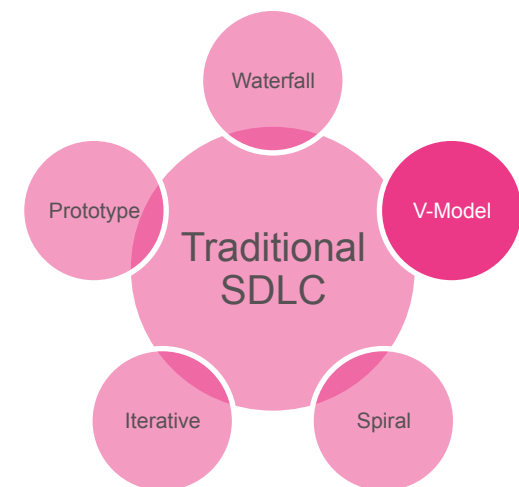


Overview of SDLC Models

Traditional – V-Model

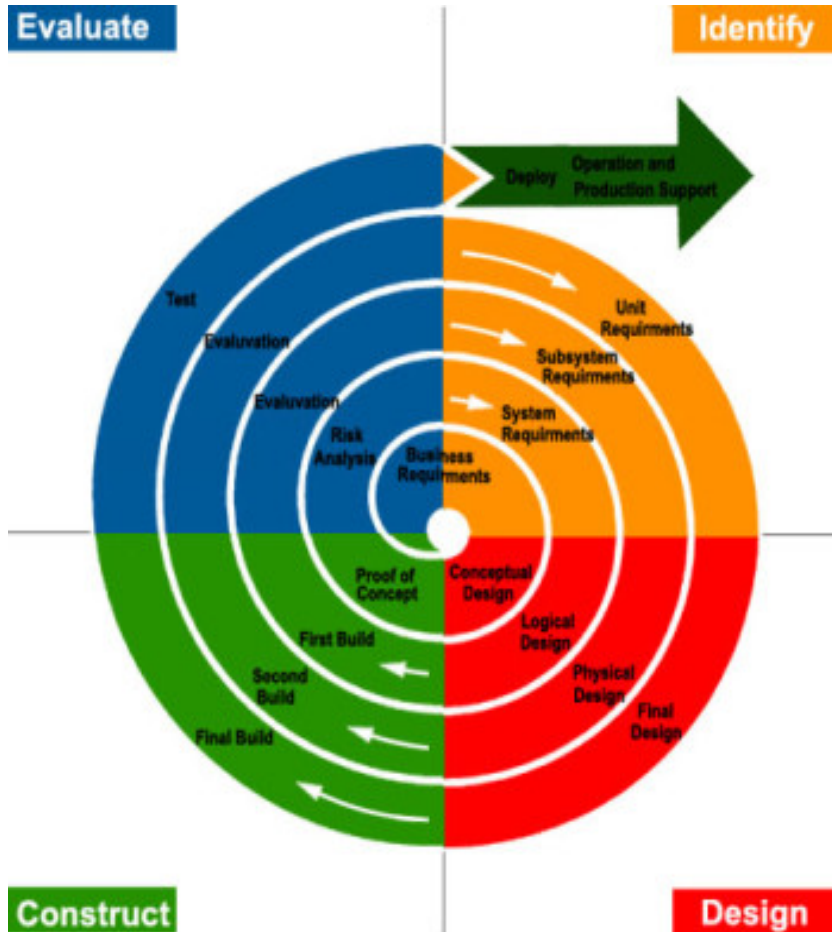


V-Model is an extension of Waterfall model and is also known as Verification and Validation Model. Execution path is sequential as in Waterfall model. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development in V-model. Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence, higher chance of success over the waterfall model

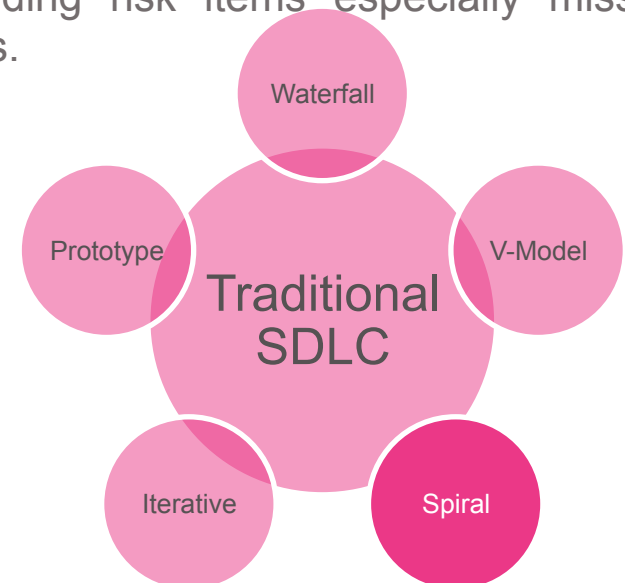


Overview of SDLC Models

Traditional - Spiral

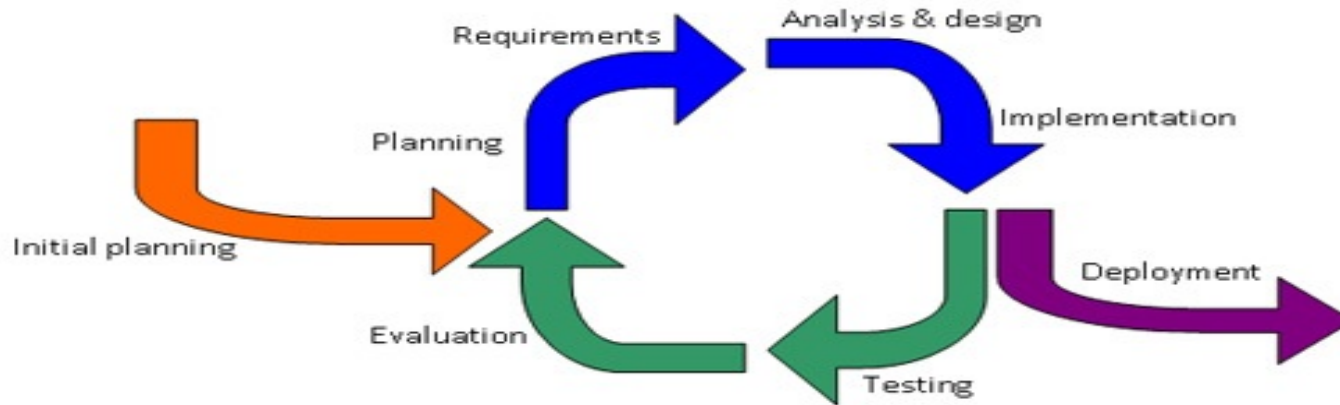


One of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement. This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on. Spiral Model takes care of risk analysis as part of its primary phase and tries to assist in avoiding risk items especially mission-critical projects.

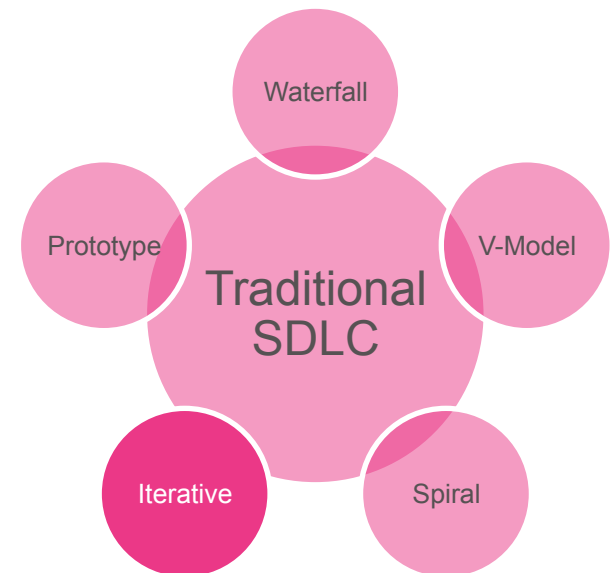


Overview of SDLC Models

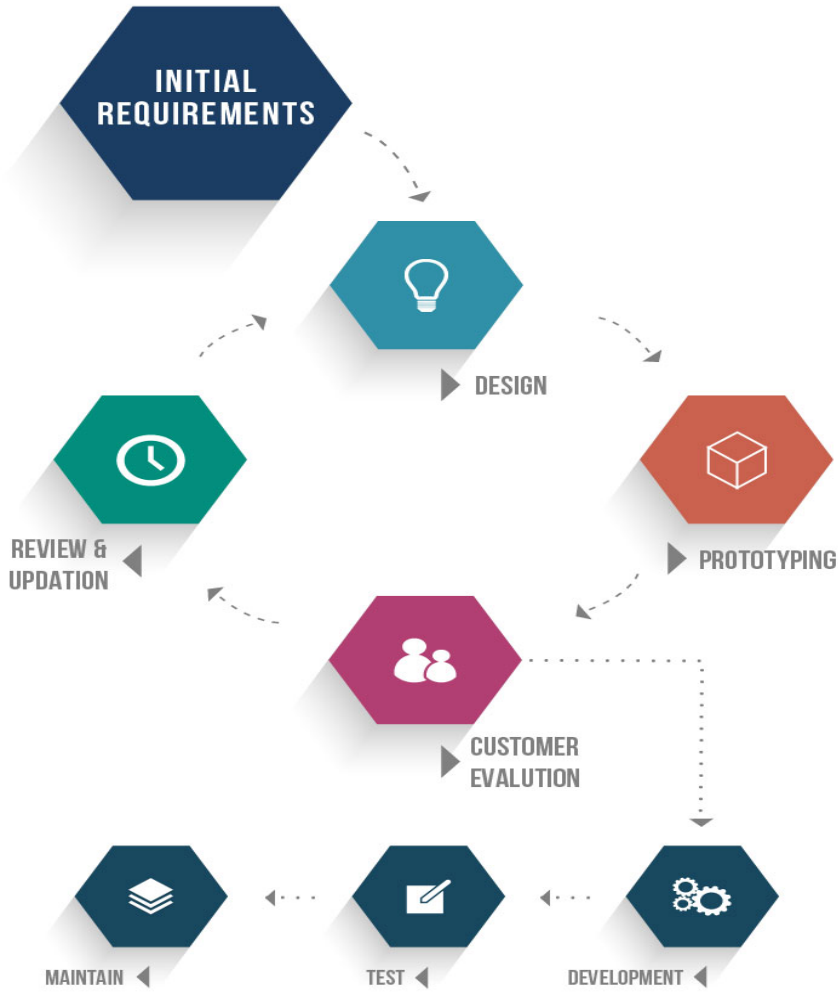
Traditional - Iterative



Iterative Model enables the users to develop the system with partial specification and then proceed with further development once the next level of specifications is refined and available. This process is then repeated, producing a new version of software for each of the cycle. The working version of the system is available early in the process and makes it less expensive to implement changes..



Traditional - Prototype



A diagram illustrating the Traditional SDLC (Software Development Life Cycle) models. A central light pink circle is labeled "Traditional SDLC". Surrounding this central circle are five smaller circles, each representing a different model: "Waterfall" (top), "V-Model" (top-right), "Spiral" (bottom-right), "Iterative" (bottom-left), and "Prototype" (left). The "Prototype" circle is a darker shade of pink compared to the others.



Overview of Agile Software Development

Overview of Agile Software Development

Agile Manifesto



Customer Satisfaction as early



Accommodate Changes



Frequent delivery of working software



Collaboration between business stakeholder and developers



Support, trust, and motivate



Enable face-to-face interactions



Working software is the primary measure of progress



Agile processes to support a consistent development pace



Attention to technical detail and design enhances agility



Simplicity

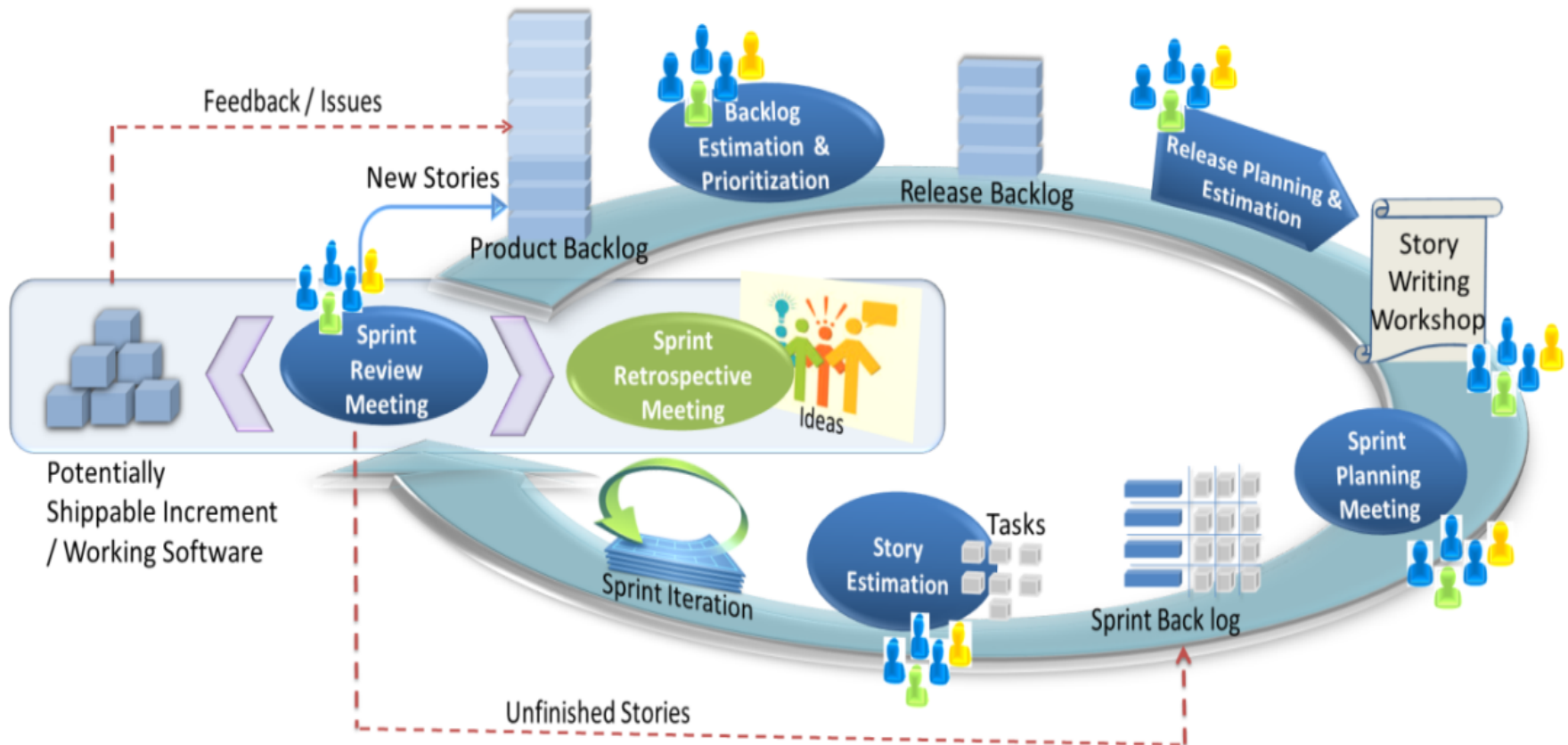


Self-organizing teams



Regular reflections on how to become more effective

PRA Agile Project Cycle





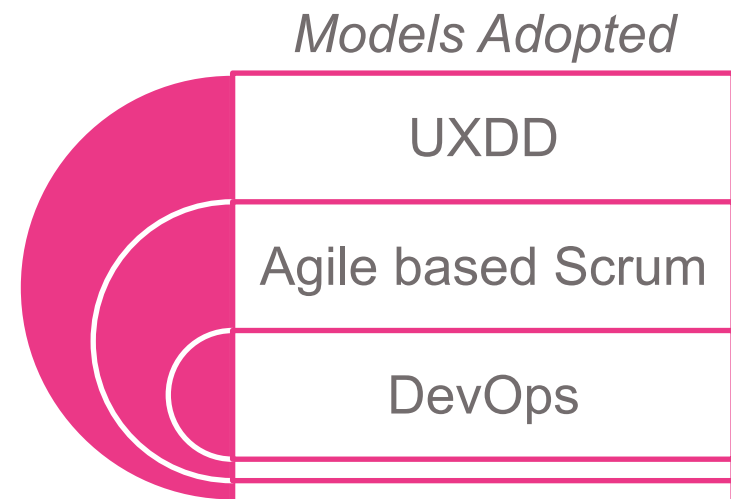
CASE STUDY - I

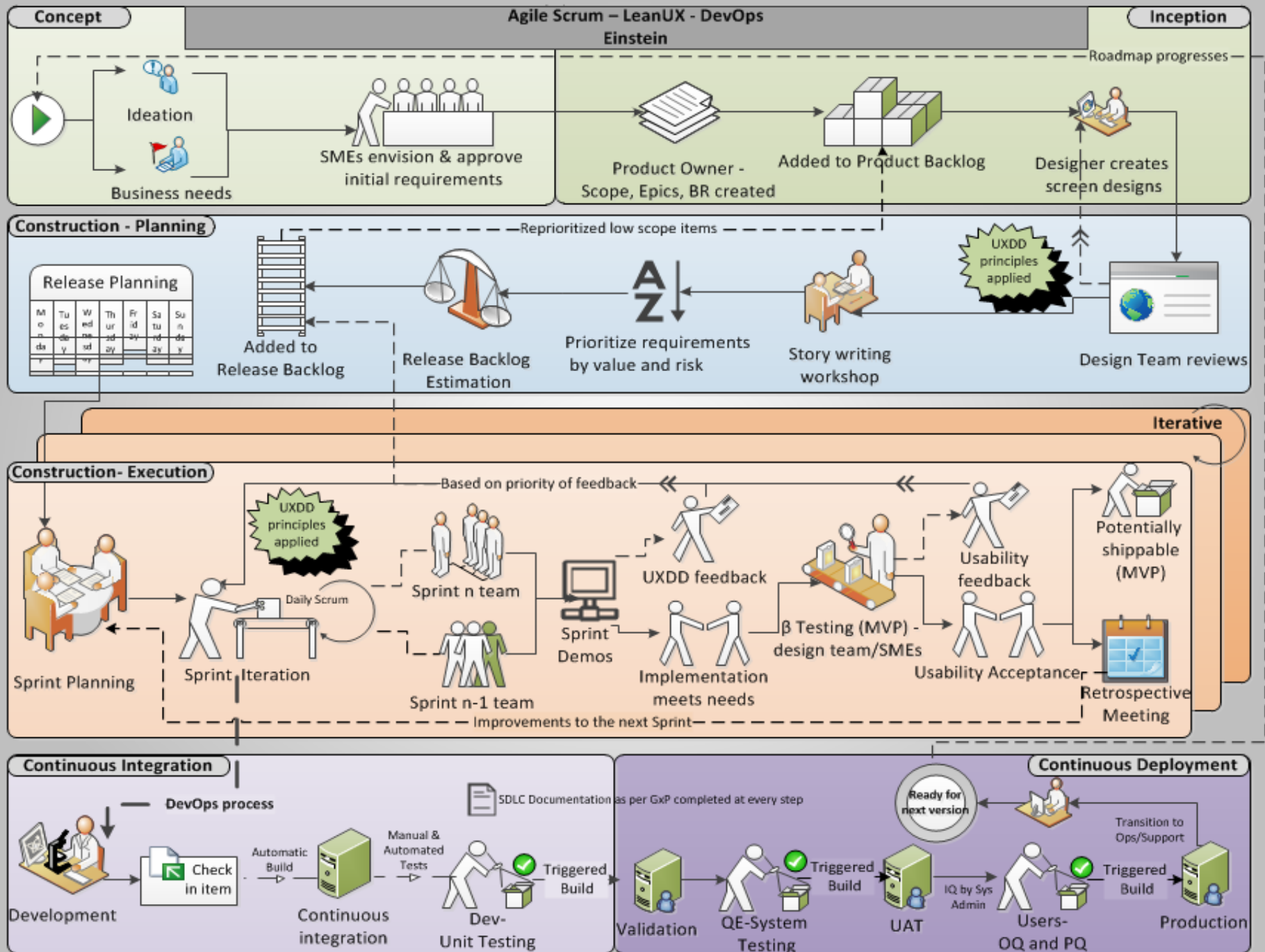
Clinical SAS Application Development in SAS GRID Environment

Clinical SAS® based Application Development is designed to handle the following business capabilities:

- ❑ Extraction of Clinical Trial Data from different sources
- ❑ Data Standardization and generation of Submission Data Standards as per CDISC Guidelines
- ❑ Creation of Analysis datasets as per Analysis Data Model
- ❑ Analysis & Reporting (Creation of Tables, Listing, Figures)

- Usability
- Business expectations on the availability
- Continuous process improvement for our internal business





DevOps Infrastructure

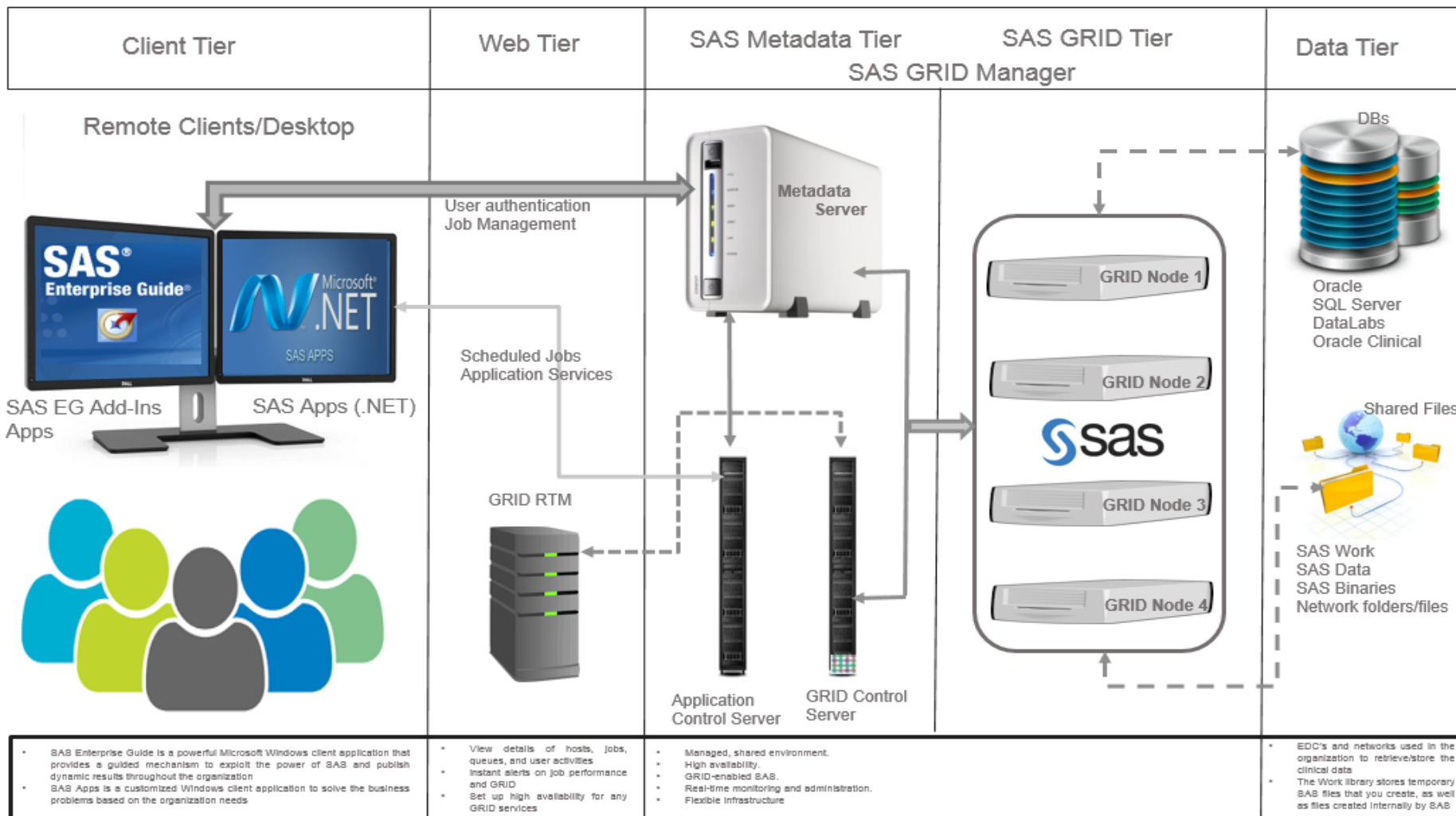
Tools



Requirements	Development (CI)	Validation (CI & CD)	Production (CD)
Language			
SAS®	SAS 9.4	SAS 9.4	SAS 9.4
	SAS Enterprise Guide 6.1 or later	SAS Enterprise Guide 6.1 or later	SAS Enterprise Guide 6.1 or later
	SAS Integration Technologies	SAS Integration Technologies	SAS Integration Technologies
C#	.NET Framework 4.6.1	.NET Framework 4.6.1	.NET Framework 4.6.1
	MS Visual Studio 2015		
	C# 6.0		
Tools	MS Team Foundation Server 2015	MS Test Manager 2015	Jenkins V1.642.1
	MS Test Manager 2015	Jenkins V1.642.1	Microsoft Office Tools 2013
	Oracle 11g	Oracle 11g	Oracle 11g
	Jenkins V1.642.1	SAP Crystal Report for VS2015	SAP Crystal Report for VS2015
	SAP Crystal Report for VS2015	SAS ASSIST 9.4	Internet Information Services (IIS) Manager
	Beyond Compare 3.3.12	Microsoft Office Tools 2013	
	Microsoft Office Tools 2013	Internet Information Services (IIS) Manager	
	Internet Information Services (IIS) Manager		
Automation Testing	Unified Functional Testing 12.0.1	Unified Functional Testing 12.0.1	
	Jenkins V1.642.1	Jenkins V1.642.1	

Deployment Environment

SAS GRID





CASE STUDY - II

Custom Task Development & Deployment in SAS

Develop Custom Task Using Microsoft .NET for SAS EG such as,

- ☐ Create a UI as a front end for a SAS procedure
- ☐ Create a interface to SAS macro

Model Adopted



Agile based Development &
Deployment



Benefits



Collaborative and Interdisciplinary



Faster time to Market



Higher Quality



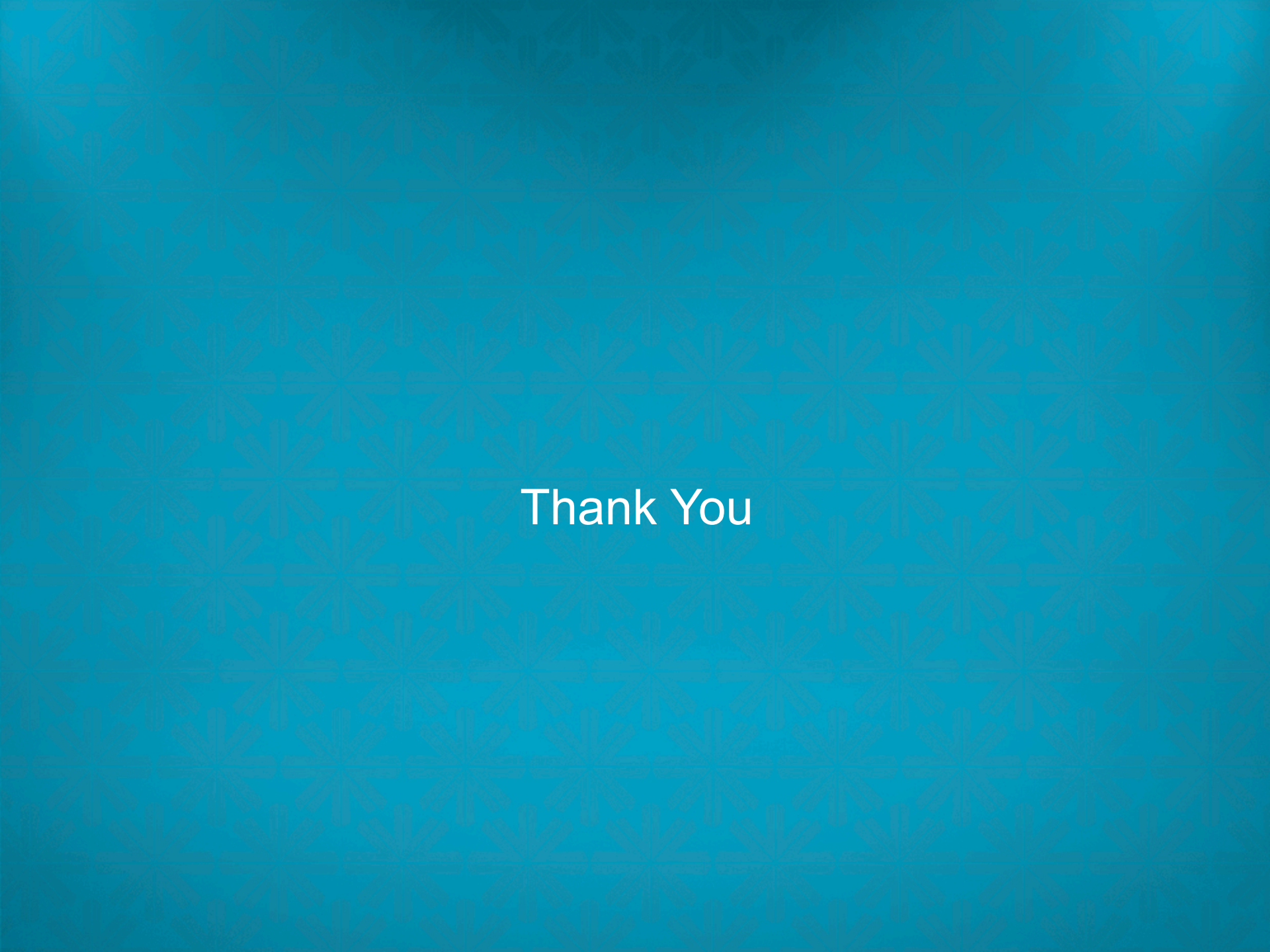
Better Products



Lower Costs



Happier Teams



Thank You