Agile estimation challenges

when starting a new team for a new product

May 2019

Eric van der Vliet

Agenda

CGI

Experience the commitment®

- The context
- Why Agile
- What is Agile?
- Challenges
- Conclusions

ICEAA

Experience the commitment®

The context

The context

- Team
  - A new scrum team
  - Experienced developers
  - Experienced scrum master

- Product
  - A new web based product
  - Requirements document in Features and User Stories

- The platform
  - Microsoft .NET Core / C#
  - Micro service architecture

- The development environment
  - Continuous Integration/Continuous Deployment (CI/CD)
  - Static code analysis
  - Automated testing

CGI

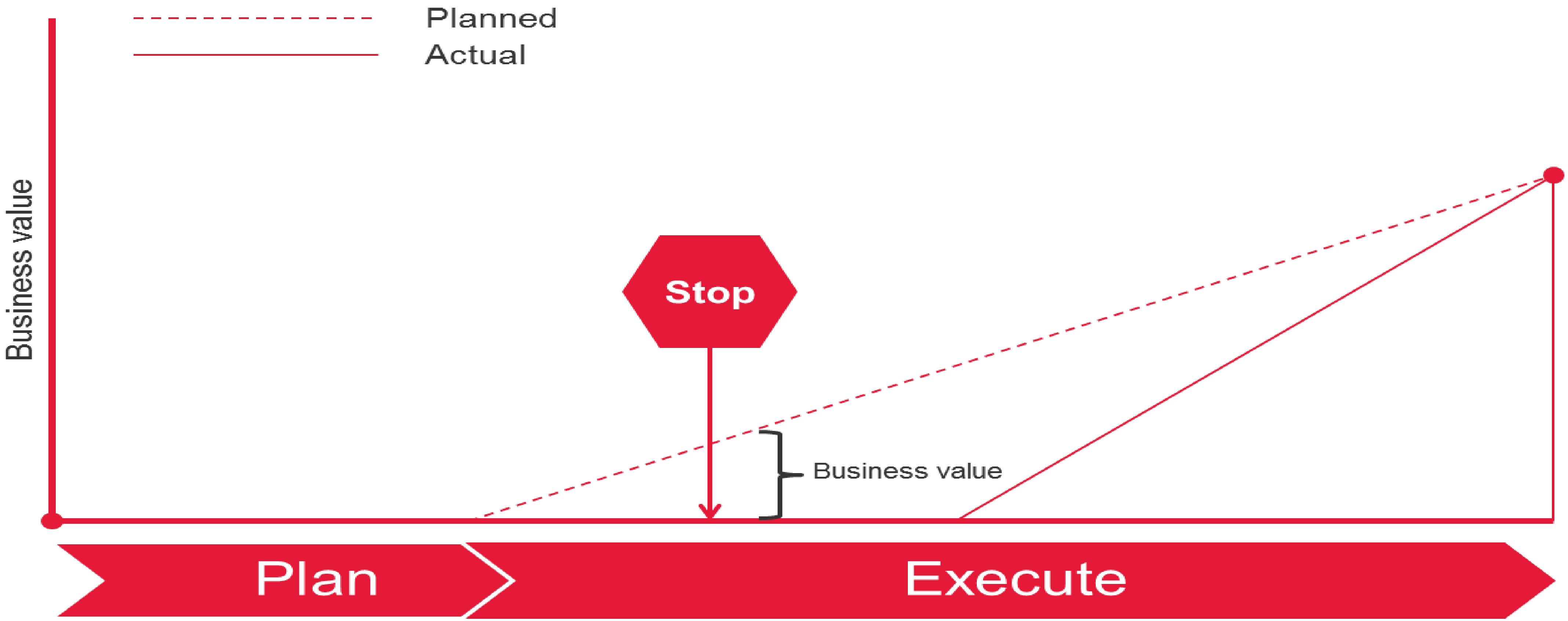
Experience the commitment®

Why Agile?

Traditional lifecycle

- Concept of the solution has been defined
- Functionality has been defined in users stories
- A Big Design Up Front (BDUF) will be challenging
- First results will be shown after 3-6 months resulting in a large amount of changes
- Validation of the system functionality / behavior only at the end of the release (acceptance testing)

Traditional lifecycles

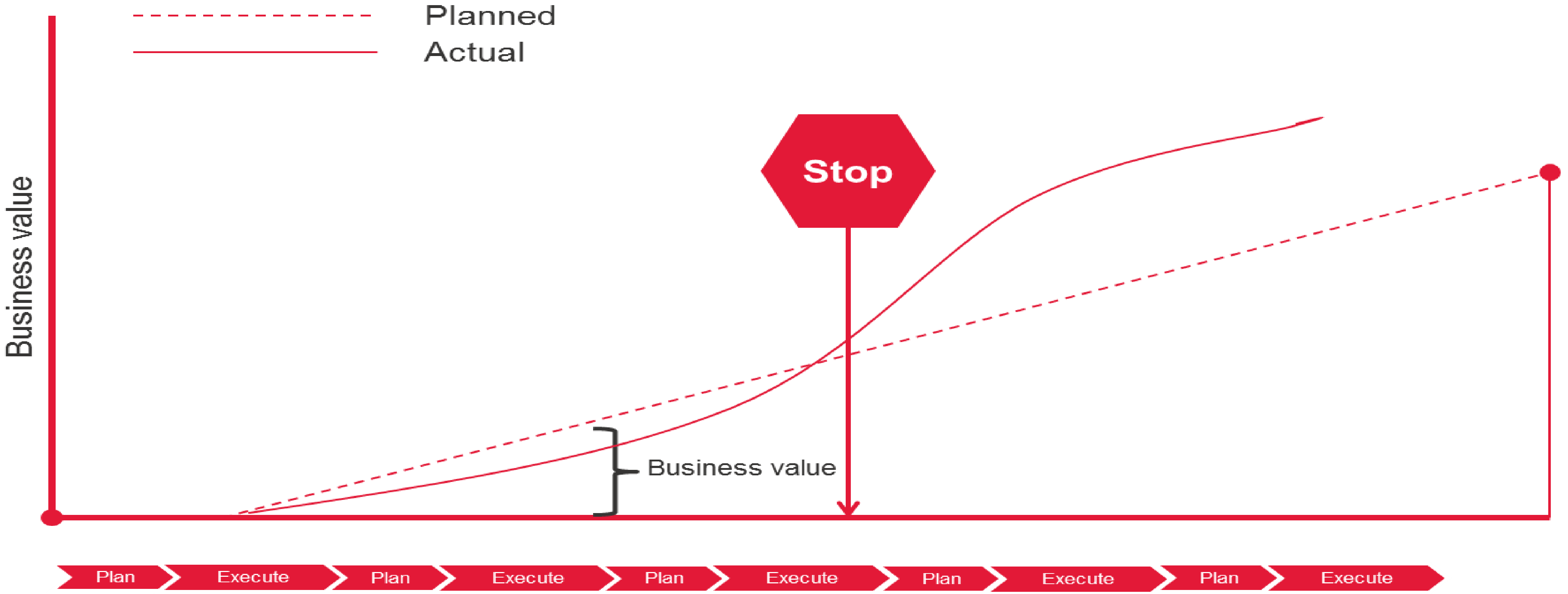


Agile lifecycle

- Controlling specifications are defined and refined during the lifecycle
- Continuous collaboration and interaction between team, product owner and end users
- Continuous validation of the system functionality / behavior (sprint level)
- Iteration level releases help the users to understand what is possible

- Budget defined; flexibility to (ex)change the requirements
- The scope can change over time based on the end user feedback

Agile lifecycles



CGI

Experience the commitment®

What is Agile?

What is Agile?

- Agile is a philosophy

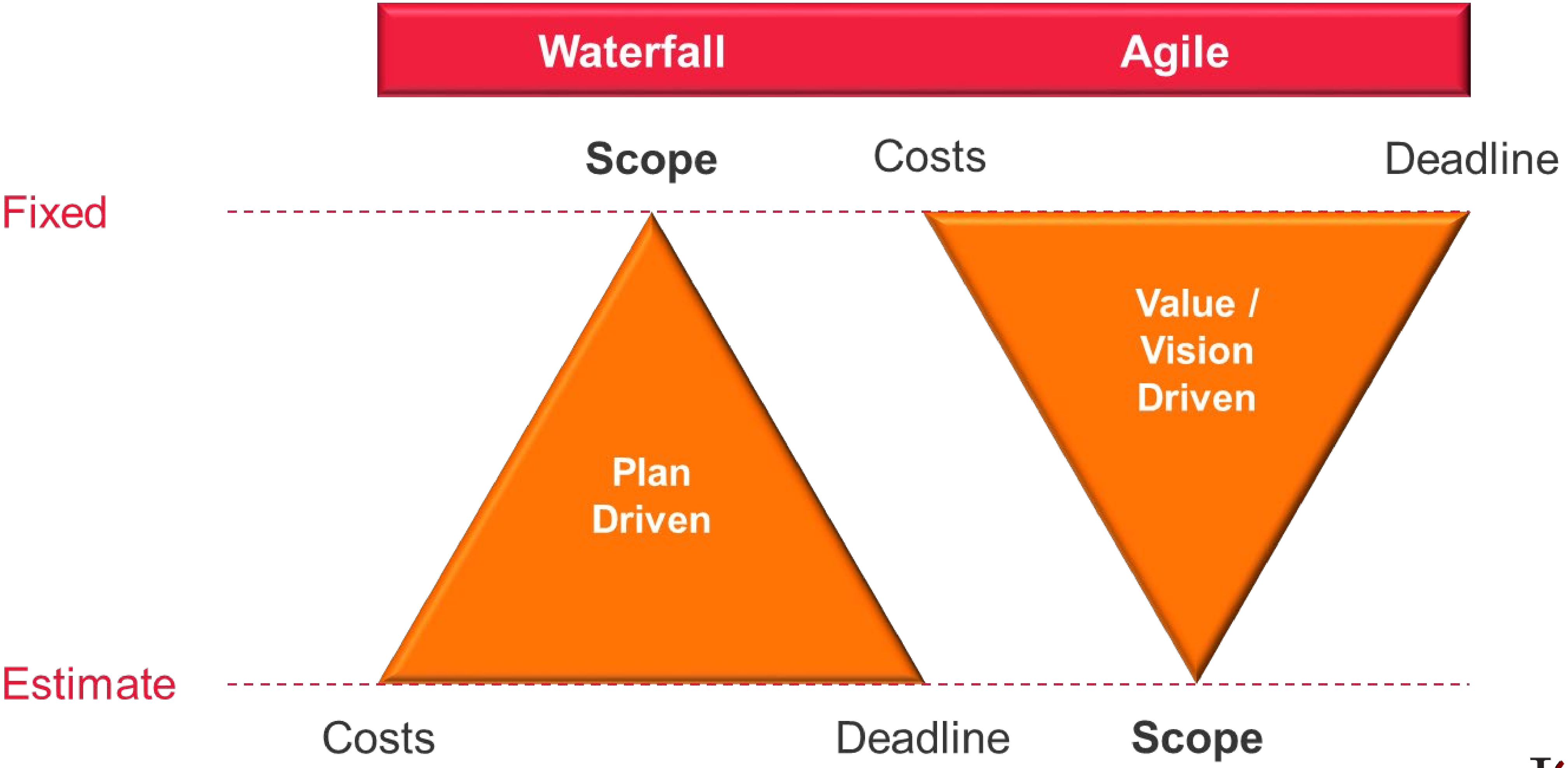
- A way of organizing yourself to achieve flexibility with respect to business objectives
- Agility means anticipation on change and focusing on value
- Flexibility is required due to a constantly, rapidly changing world around us

- Agile is not Scrum or Kanban

- Moving from a traditional way of working to an Agile way of working requires a fundamental change of:
  - Culture
  - Behavior
  - Attitude

- Agile development relies on traditional software development fundamentals
  - just approaches them to focus on value

Agile deliveries are value driven



The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over **Process and tools**

Working Software over **Comprehensive documentation**

client collaboration over **Contract negotiation**

Responding to change over **Following a plan**

While there is **value** in the items on the **right**, we **value** the items on the **left** more

Experience the commitment®

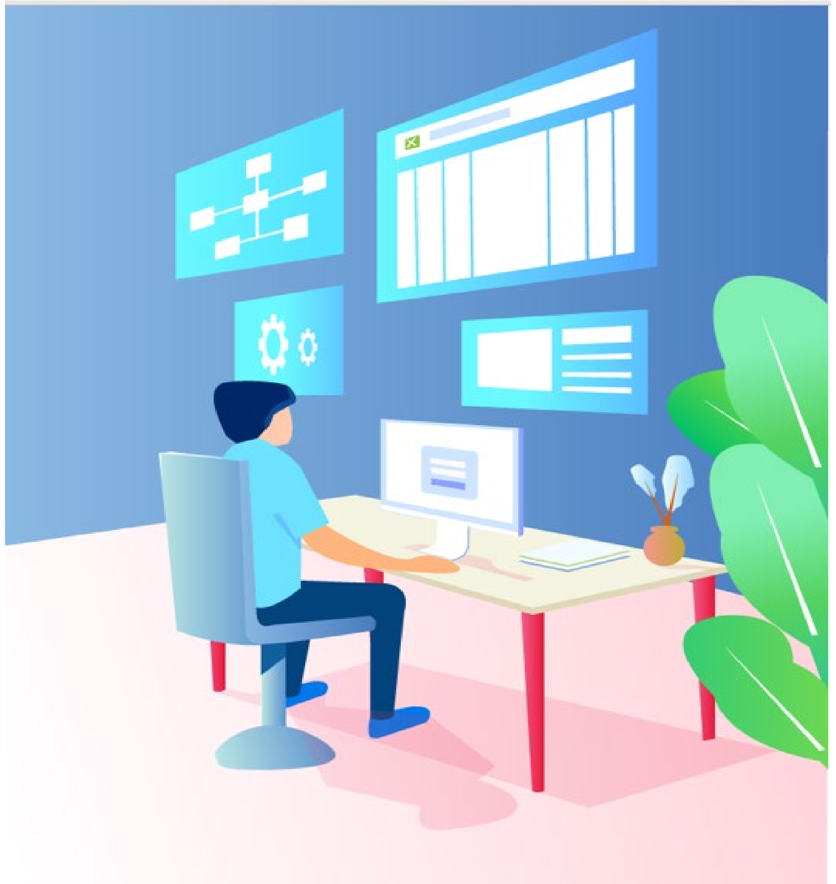
The challenges

1. The team

- User Experience (UX) -Team
  - UX Design
  - UX development

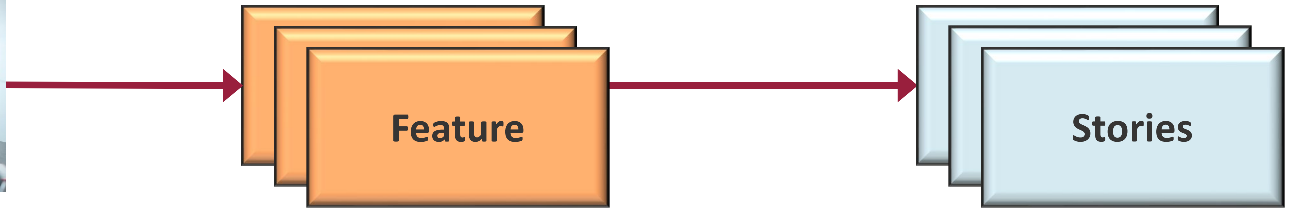
- Development team
  - Architectural design
  - Application design
  - Application development

- Infrastructure team
  - Infrastructure design
  - Infrastructure development

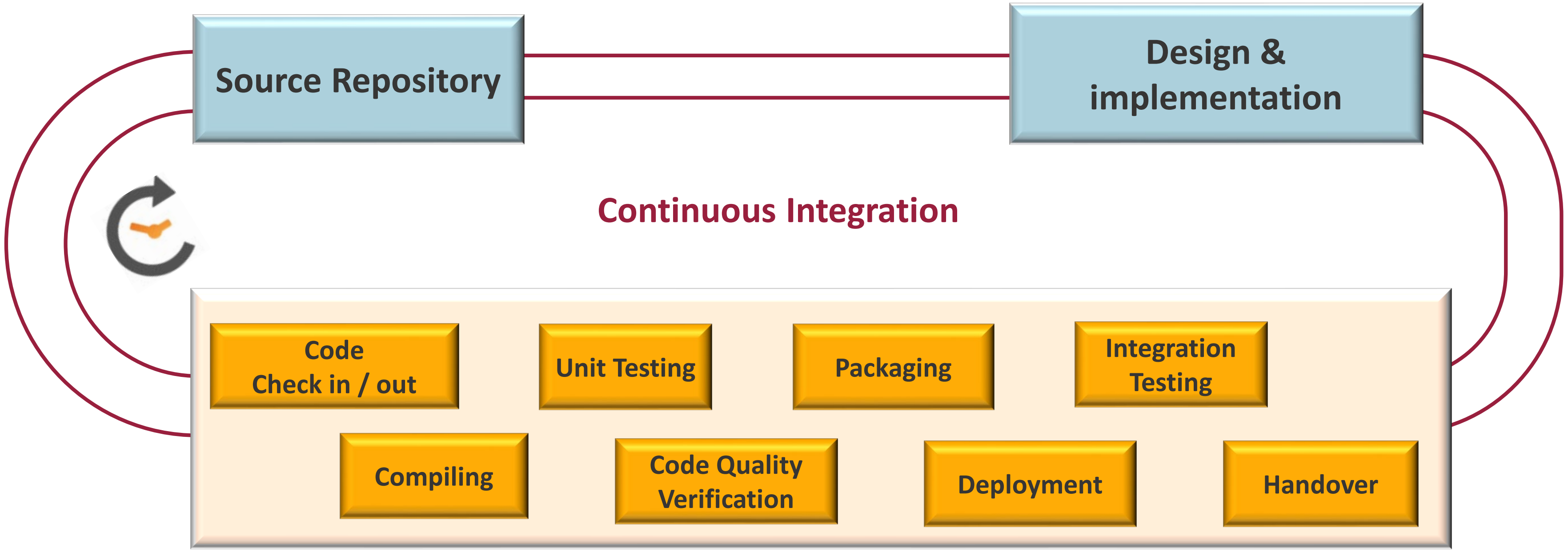


1. The team - Process and tools

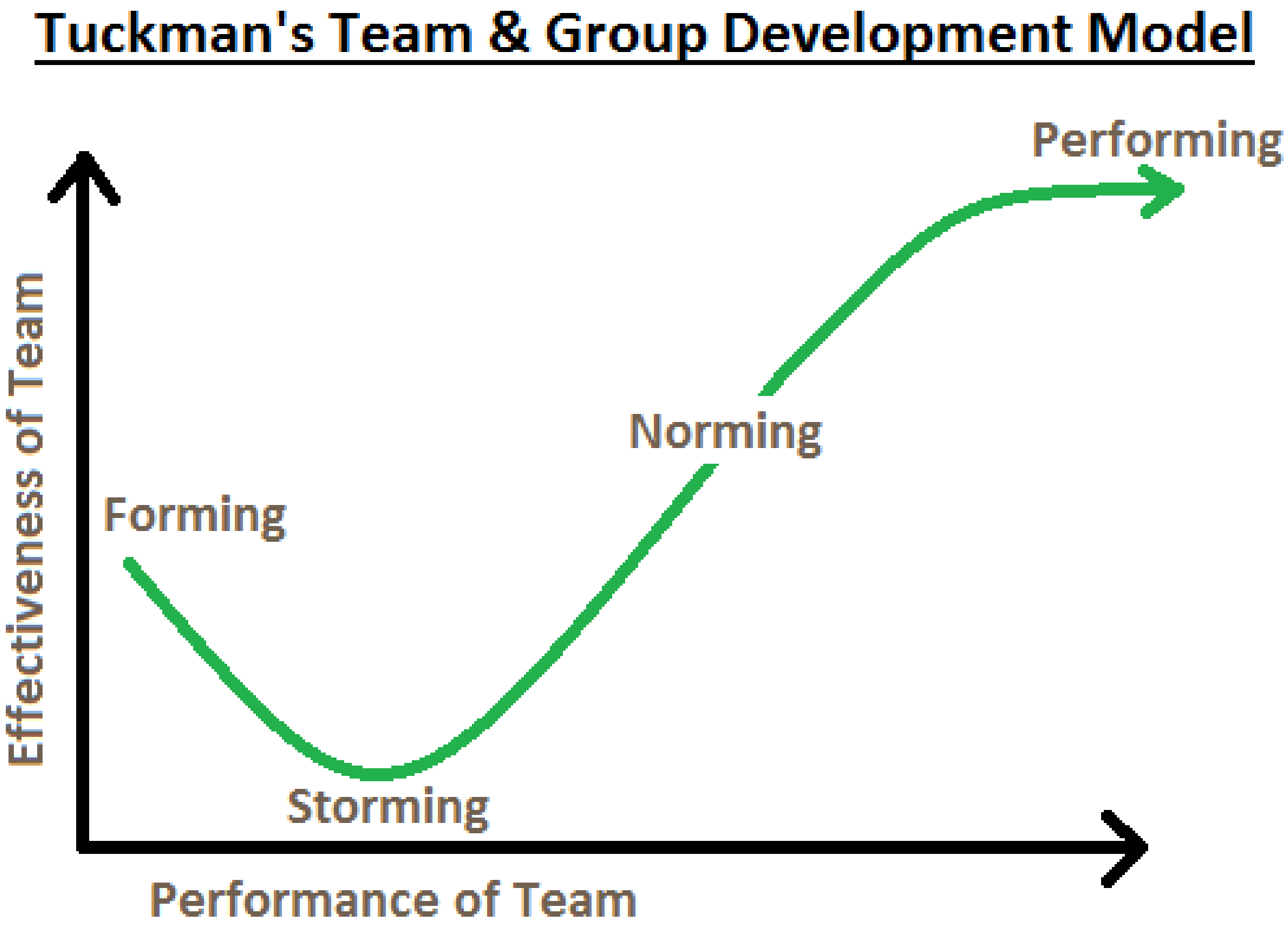
Business



- Collaboration with Product Owner (PO)
- Assumptions
- Clarification
- Acceptance criteria

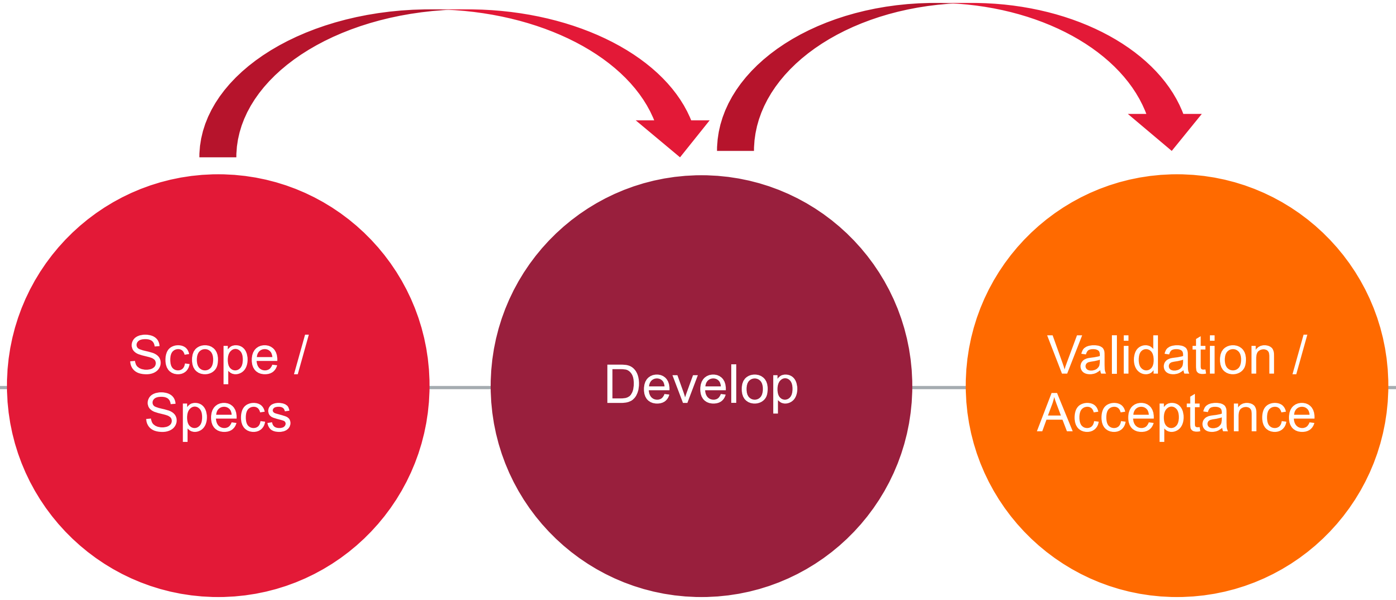


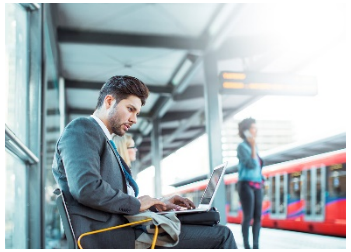
1. The team - Optimal team performance requires time



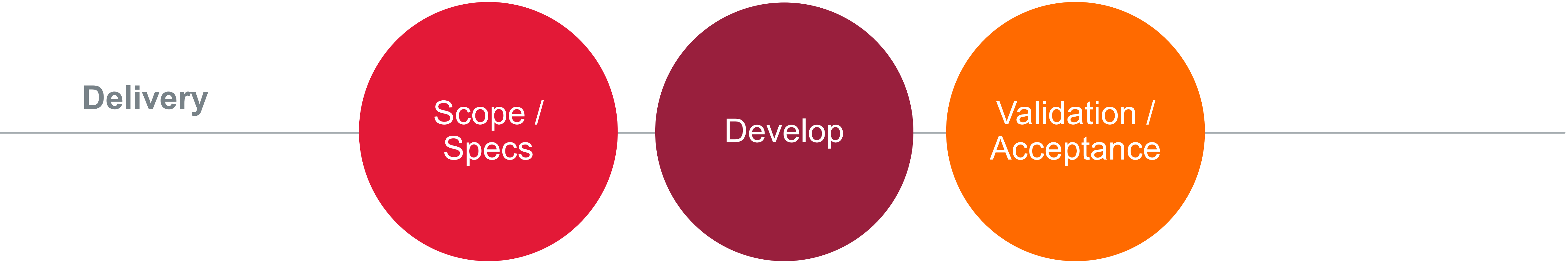
2. Communication

Delivery





2. Communication - Collaboration & interaction



3. Estimation

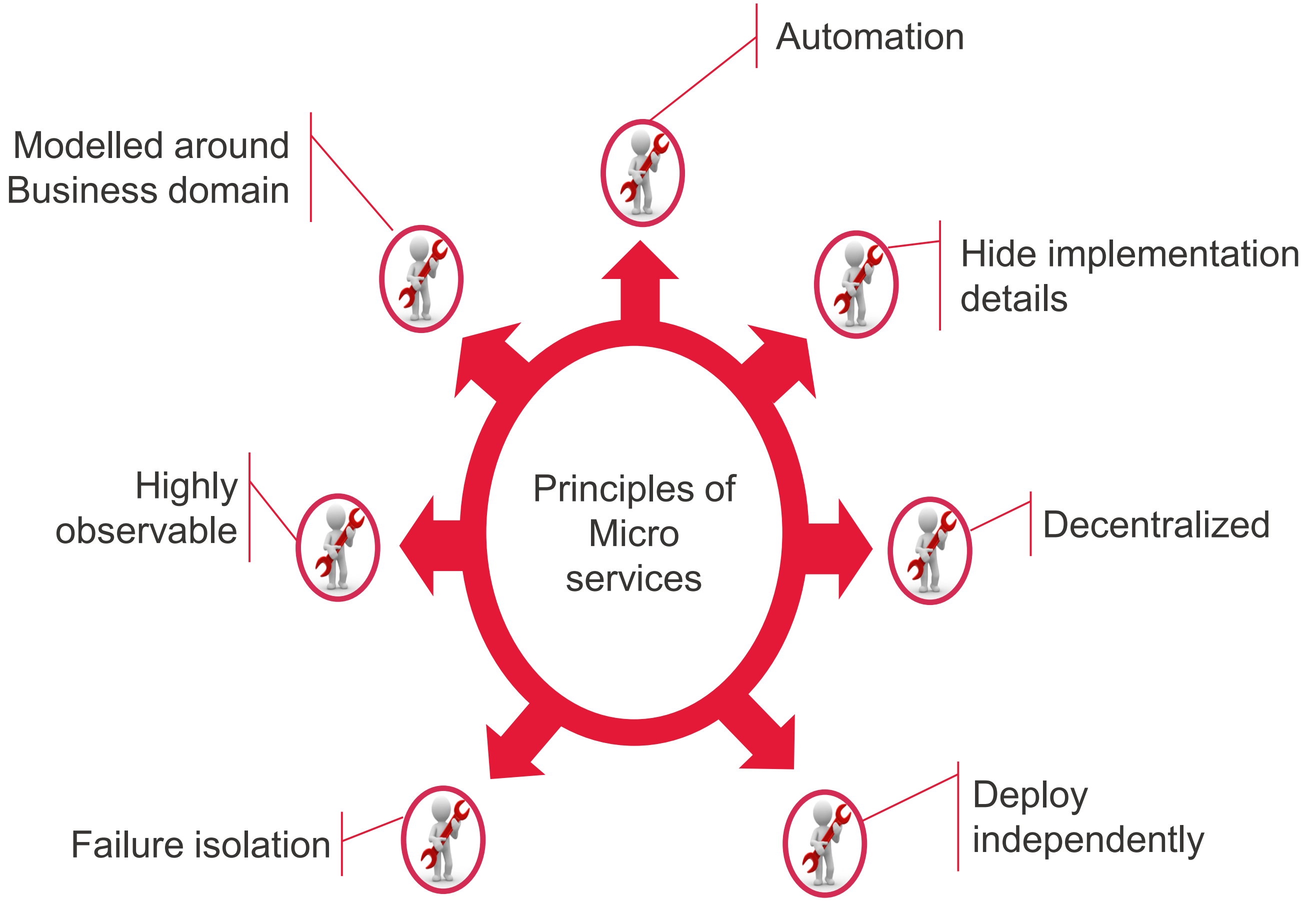
- Scope has been defined but
  - What will fit in a sprint?
  - What will fit in an increment?

- Sizing of the scope
  - Features are on a high level
  - Stories are one liners; *As a <role> I want <…..> such that <…...>*

- Size has been defined but
  - Size in Function Points (FP's) and Story Points (SP's)
  - What will be the team productivity (hr / FP's)
  - What will be the team velocity (SP / Sprint)?
  - How representative is historical data?

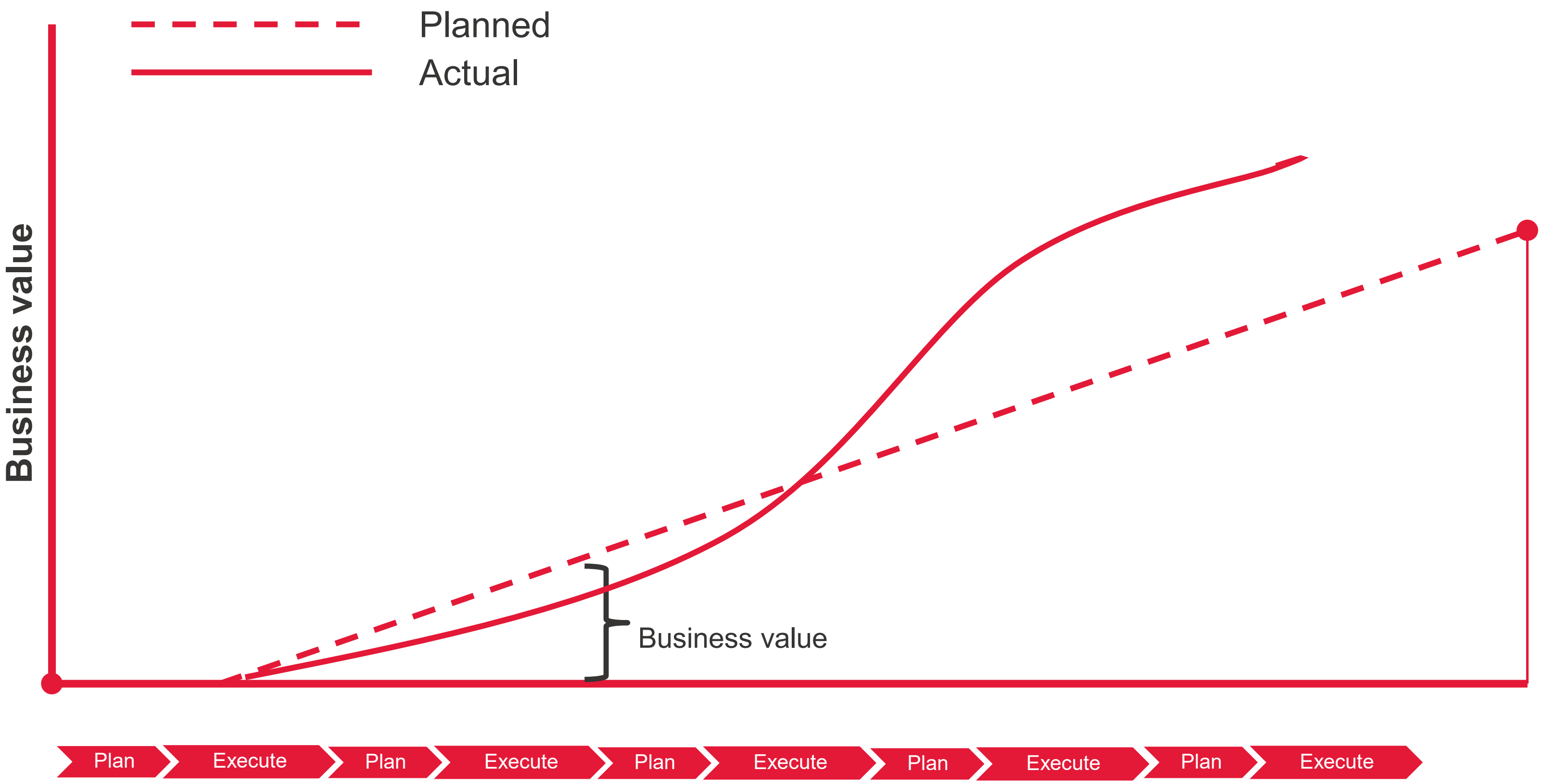
4. Architecture

- Systems made up of micro services
- Independently deployable components
- Interoperability through message-based communication

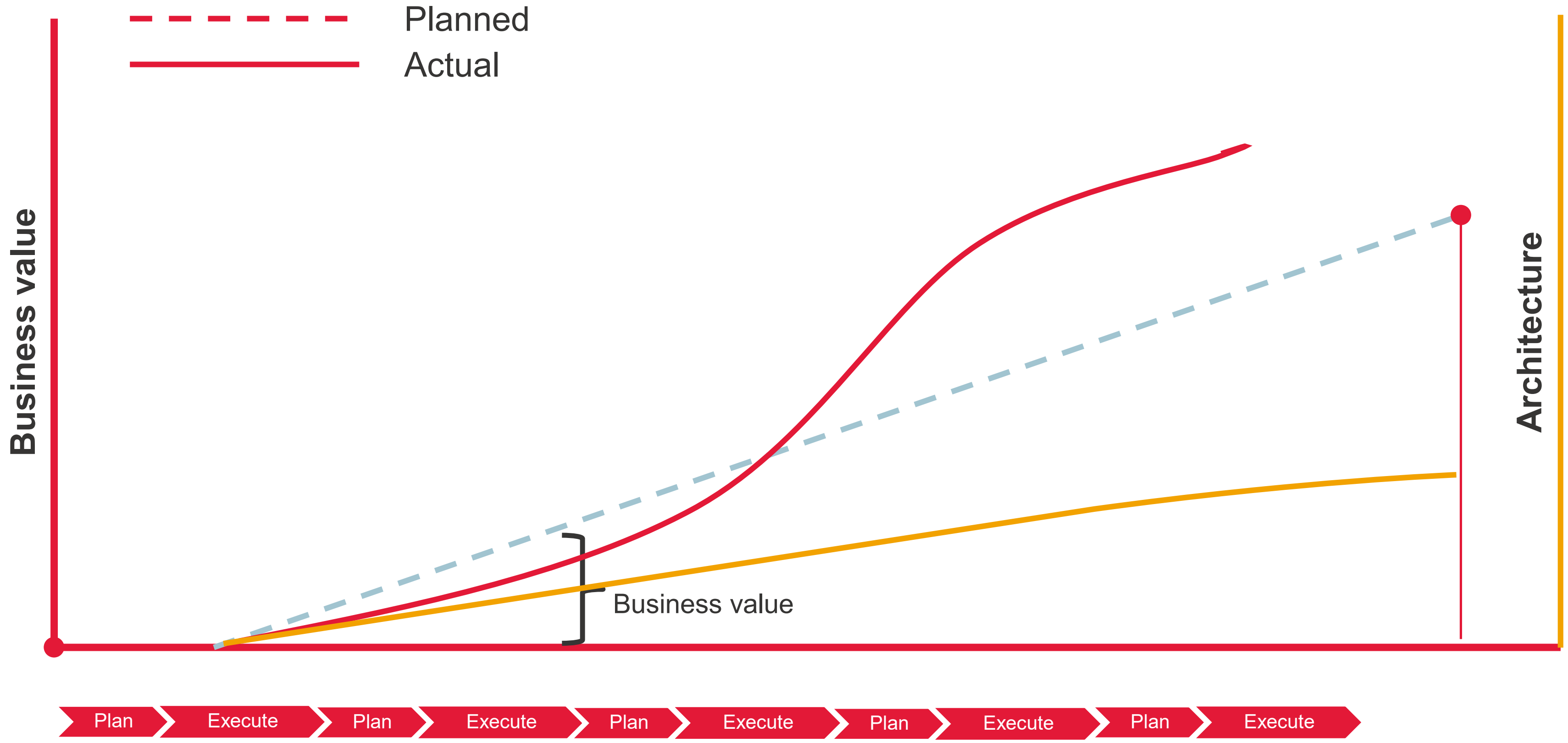

Automation
Modelled around Business domain
Hide implementation details
Principles of Micro services
Highly observable
Decentralized
Failure isolation
Deploy independently

Benefits
Increased Agility
Business meaningful capabilities
High Resilience
Greater availability, stability and scalability
Reduce Complexity
Operational Cost Saving

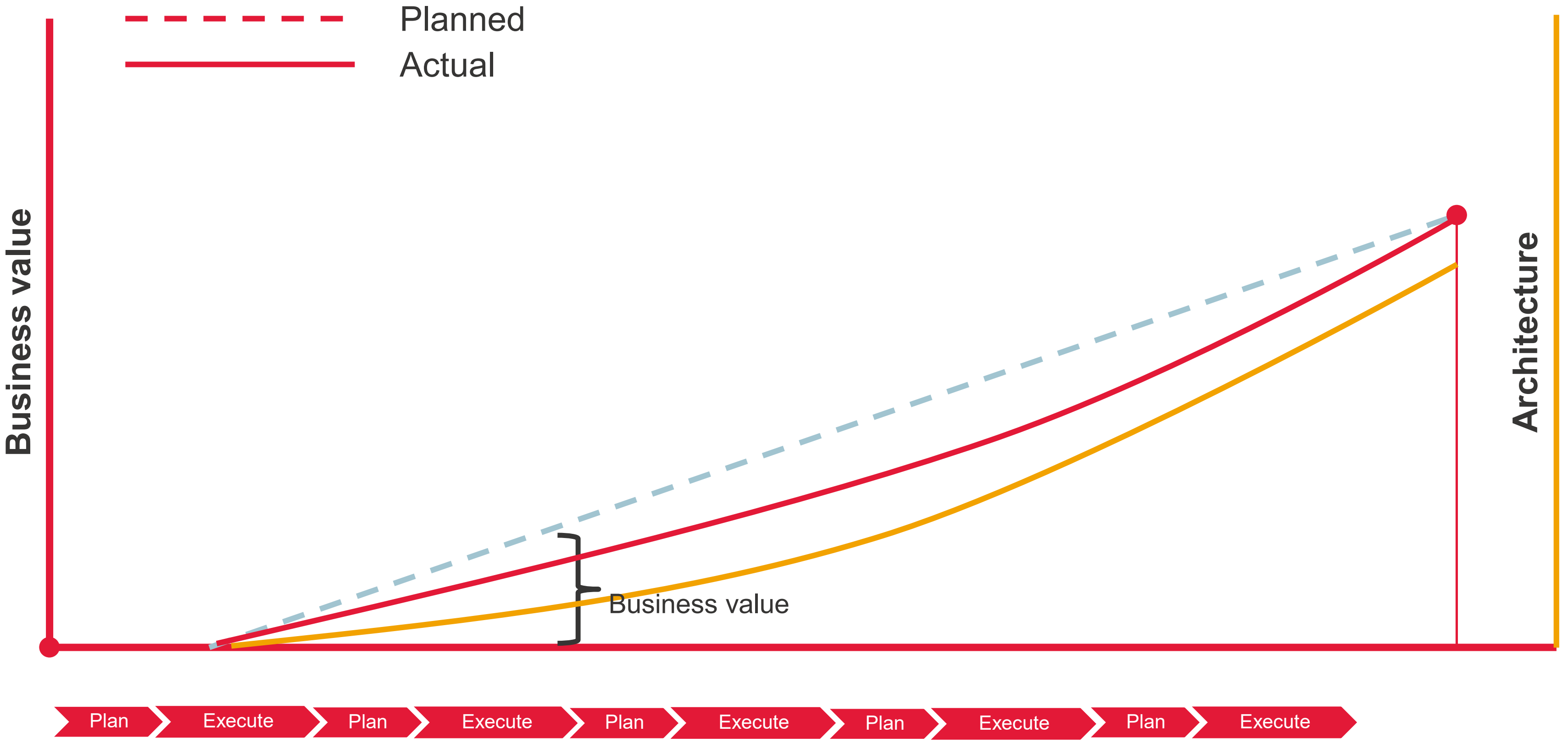
4. Architecture – Planned vs Actual Business Value



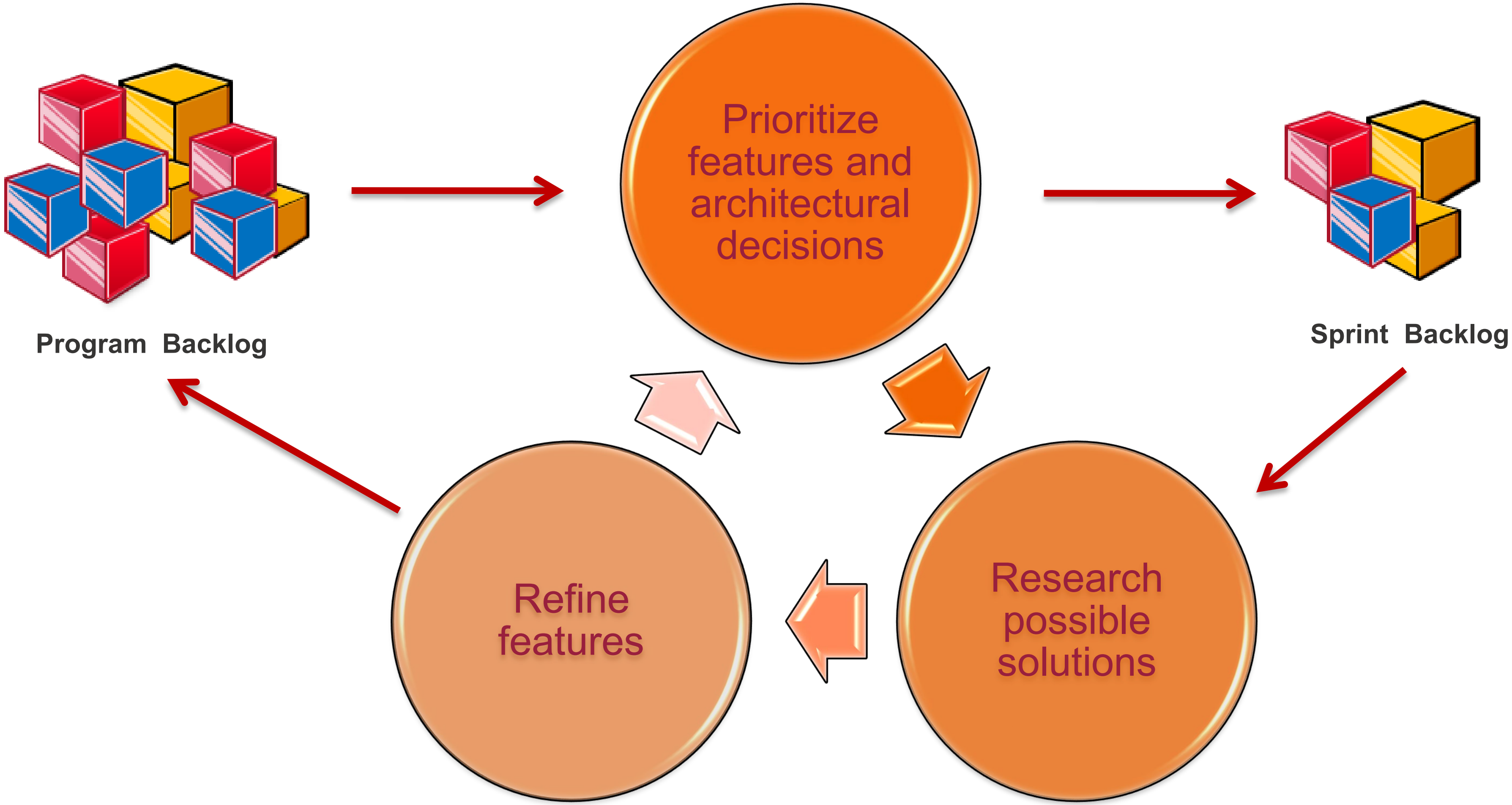
4. Architecture - Product architecture is required



4. Architecture - Business value vs Product Architecture

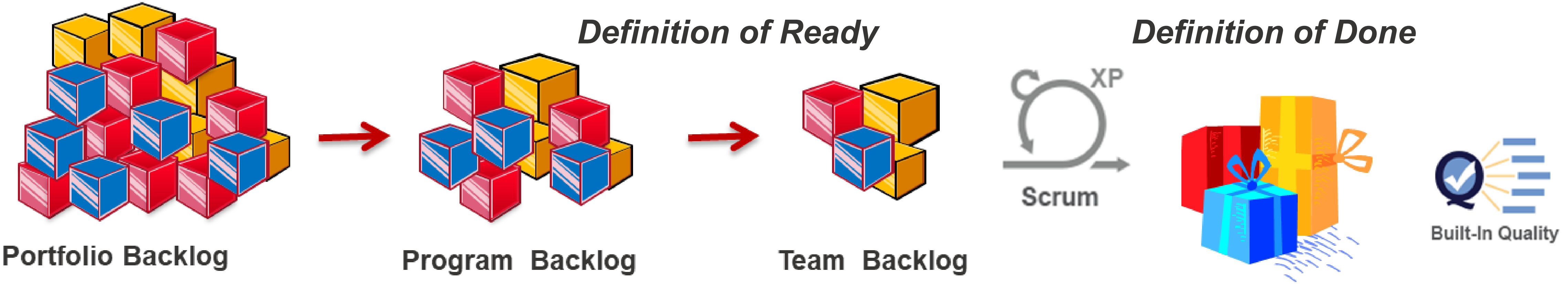


5. Product Management

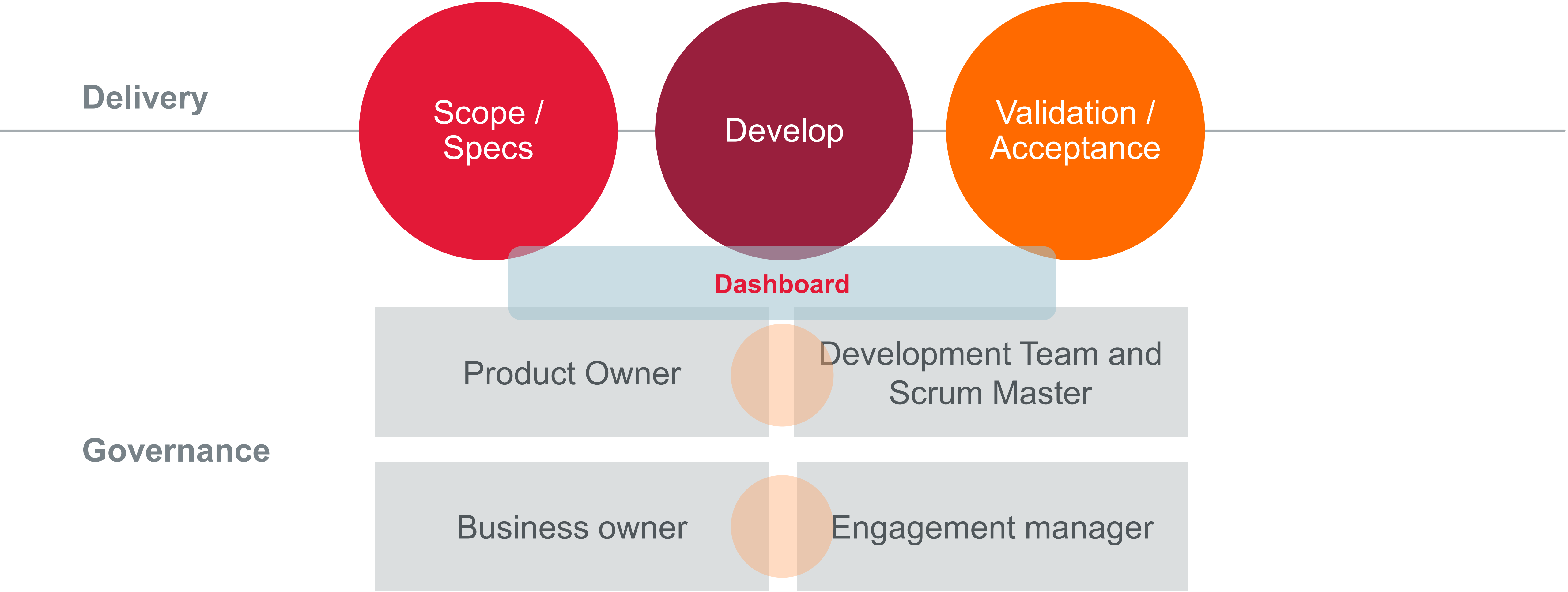

Prioritize features and architectural decisions
Program Backlog
Sprint Backlog
Refine features
Research possible solutions

5. Product Management - Team efficiency

- Team efficiency depends on the quality of the sprint backlog
- Product management is responsible for providing the stories based on the defined features
- The product owner will work with the team to clarify the user stories
- User stories must be detailed enough to be able to make an accurate estimate for the sprint
- A Definition of Ready is applied to verify if the user stories are detailed enough
- A Definition of Done is applied on the sprint results



5. Product Management - Collaboration & interaction



Delivery

Scope / Specs

Develop

Validation / Acceptance

Dashboard

Governance

Product Owner

Development Team and Scrum Master

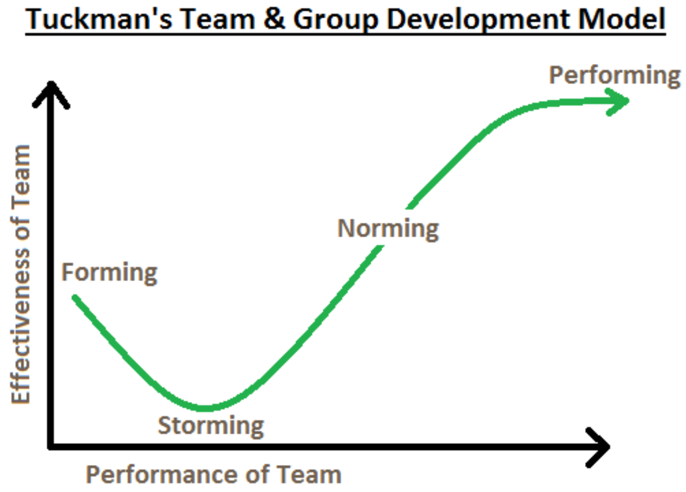
Business owner

Engagement manager

Initiation phase

A good practice is to agree on an initiation phase for 2 to 5 sprints to:

1. Build a stable **team** and establishing the minimal basis infrastructure and tooling
2. Establish the proper **collaboration and interaction** within the team
3. Establish a base line on **estimates** and expected output
4. Establish a minimal basic **architecture** to be able to deliver the requested functionality
5. Establish a **product management** process that supports efficiency of the team



CGI

Experience the commitment®

Conclusion

ICEAA

Conclusion

- Agile is used as the general term for the whole delivery but the team applies Scrum or Kanban
- Agile deliveries can be well estimated and controlled but this requires:
  - A well defined development process and supporting tools to monitor the progress and the quality
  - Familiarity of the team with the Agile development approach and culture (initiation phase)
  - Collection of metrics to be able to improve the estimation accuracy of sprints / increments
  - A proper Product Management process that provides the requirements for the next sprint / increment
  - A good architect who defines the basic architecture and evolve the architecture during the program

Questions?

Thank you!

Mail:	**eric.van.der.Vliet@cgi.com**
Linked-in:	**www.linkedin.com/in/eric-van-der-vliet-8963652/**