DB AG/Christian Bedeschinski

www.thalesgroup.com/germany

Agile Software Development compliant to Safety Standards?

Christian Scholz

Thales Transportation Systems

Security and mobility in a networked world.

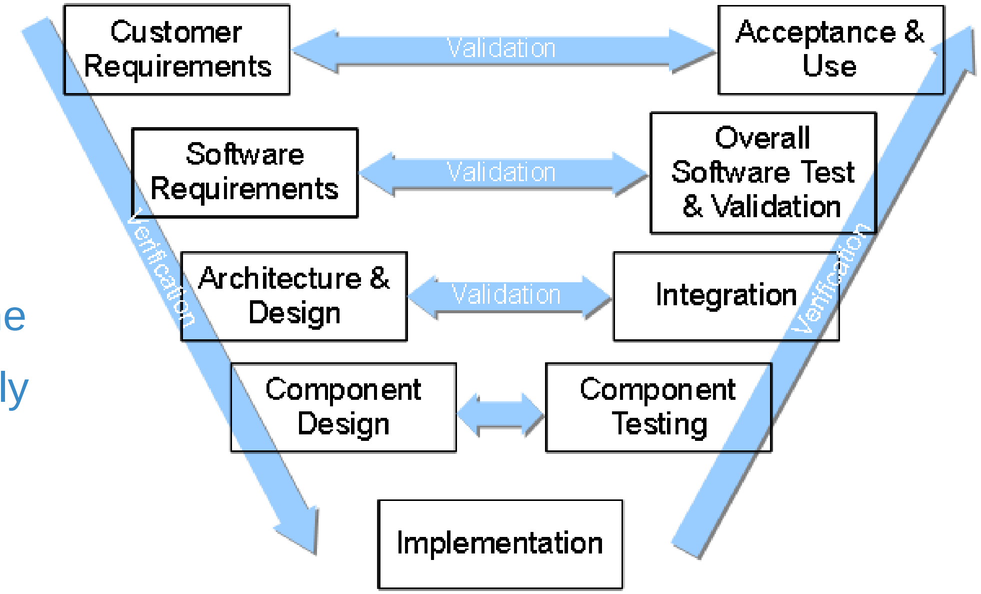
THALES

Content

- Motivation
- Agile Software Development
- Research Question
- Safety and Agile
- Experience from Thales

Motivation

- **Thales**
  - Software Development for railway applications
  - Traditional development: Waterfall, V-shaped model
    - Requirements fixed
    - Plan created
    - Sequential phases

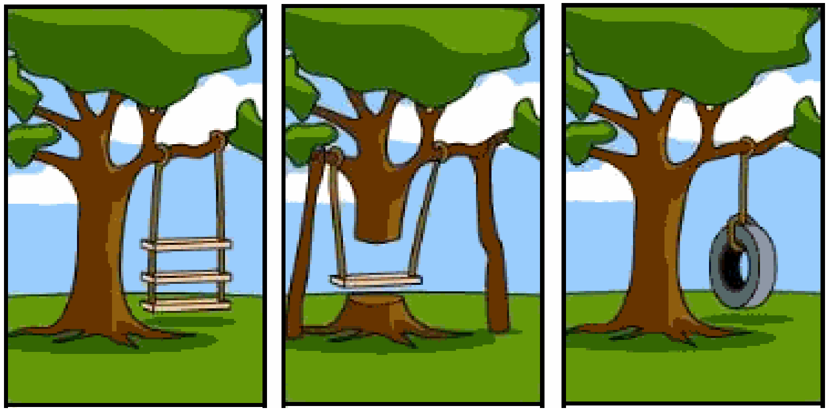


- **Problems**
  - Projects exceed budget and time
  - Software Development too costly
  - Market pressure

- **How to improve?**

Problem of late changes

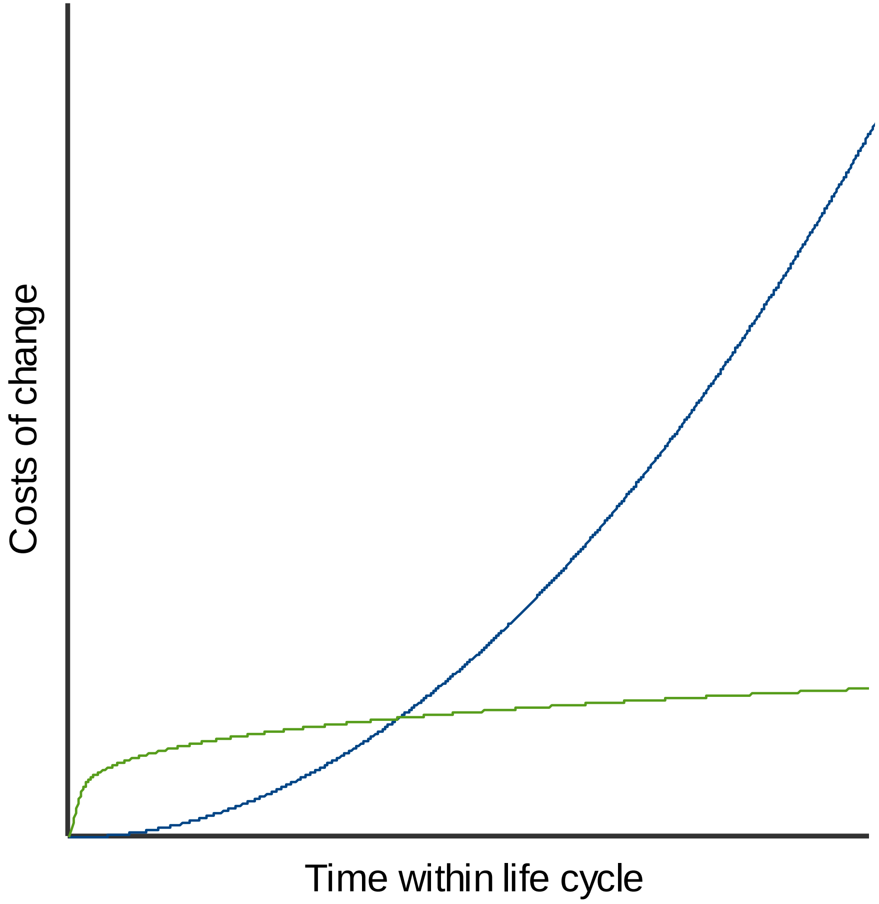
- **Problem:**



  - Customer does not know or can't explain its requirements
  - Vague, no precise and unstable requirements
  - Developed something the customer did not want

- **Result:**
- **Changes late in the life cycle**
  - Plan does not work – More costs and time

Reaction



- Traditional
  - Changes late in the life cycle - extremely costly
  - Avoid late changes
  - Build it right the first time
  - Big effort upfront
  - Many processes, detailed planning
  - ➢ Still need for late changes
- Agile Software Development
  - Accept late changes
  - Lower the costs-of-change-curve

Agile Software Development

- **Experience and best practices**
  - Industry experts and practitioners
  - Different agile methodologies
    - Scrum, eXtreme Programming (XP), Crystal, and more
  - Also roots in lean manufacturing and development

- **2001 Agile Alliance**
  - set term „Agile Software Development"

- **Manifesto for Agile Software Development**

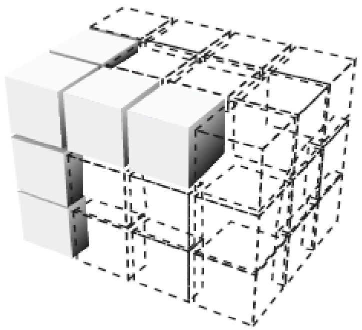
Manifesto for Agile Software Development

- **Individuals and interactions over processes and tools**
  - People centric: Adapt process to people
  - Light process
  - Management style: Leadership-and-collaboration
- **Working software over comprehensive documentation**
  - Priority on the value adding activities
- **Customer collaboration over contract negotiation**
  - Closer customer collaboration
- **Responding to change over following a plan**
  - Refine and extend requirements

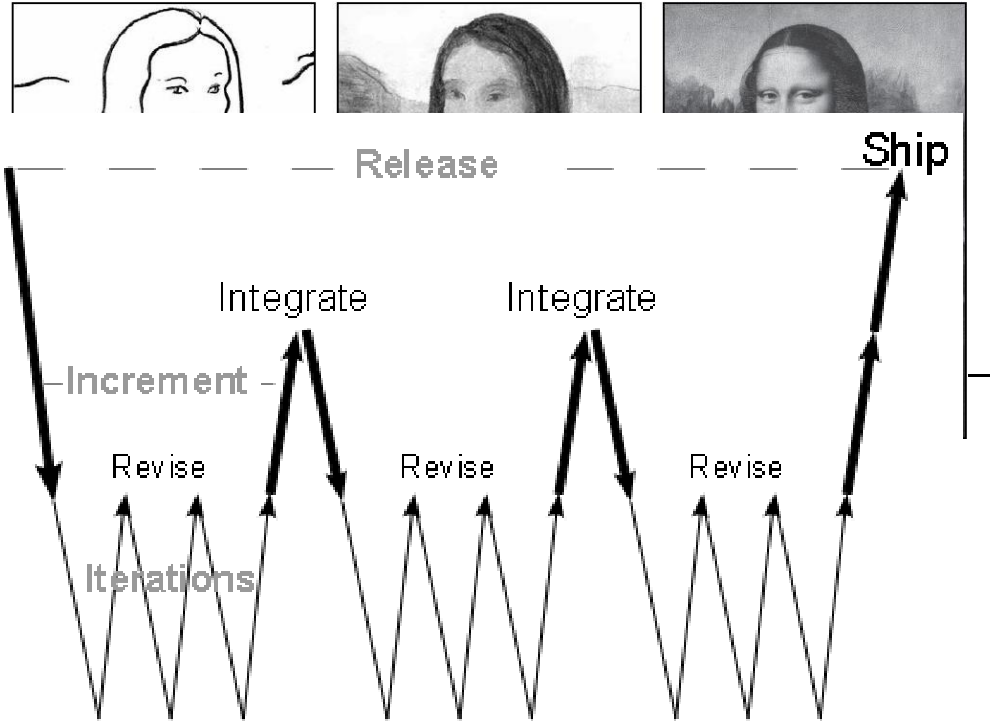
Incremental and iterative development

- **Incremental**
  - Development split into series of partial products
  - Working software earlier in the project
  - Increasing functionality with each increment

Figure 2: *Incremental Development, Stage1*

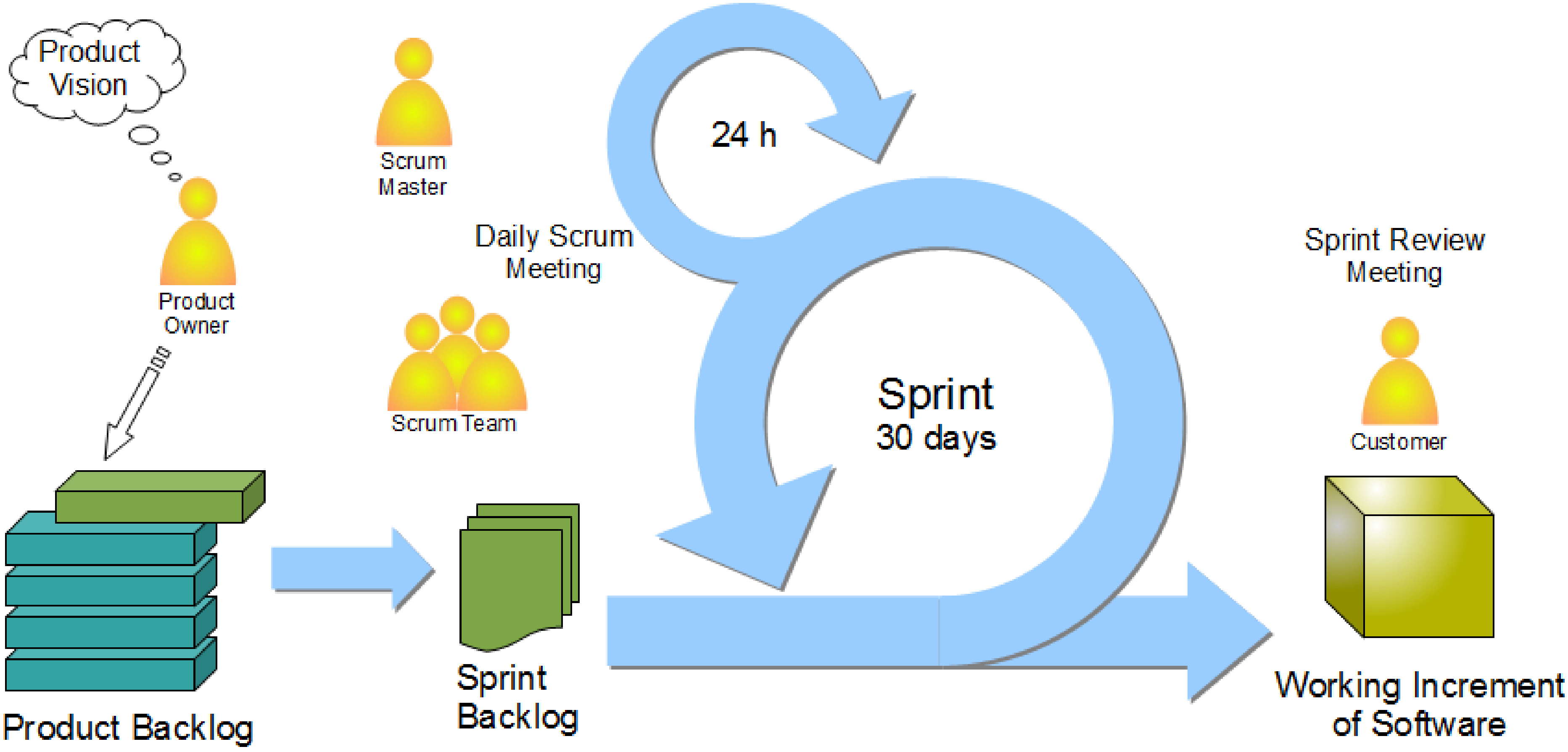


- Iterative
  - Revise strategy
  - Improve system
  - Together with customer
    - **Small V-Models**
    - **No strict sequence**



Scrum by Jeff Sutherland and Ken Schwaber

- **Framework for managing software development projects**



Research question – Master Thesis

- **Agile Software Development could help Thales**
  - Attacks existing problems
  - Success stories from other organizations

But,

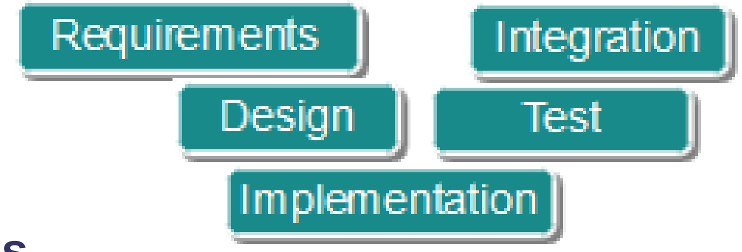
- **Safety-critical development for railway applications**
- **CENELEC standards**
  - Prerequisite for certification
  - Assumption: Standard asks for traditional V-Model

Agile Software Development – CENELEC?

Detailed study of CENELEC EN 50128

- **Goal of the standard**
  - Reduce systematic errors – down to a tolerable level
  - By setting standards on the process

- **Requires mandatory development phases**
  - Strict sequence is impractical
  - Accommodates refinement, prototyping or incremental approaches


Requirements

Integration

Design

Test

Implementation

- **Additional software assurance activities**
  - Verification, Validation and Assessment
    - Obey rules of interdependency
    - Two pairs of eyes are better than one

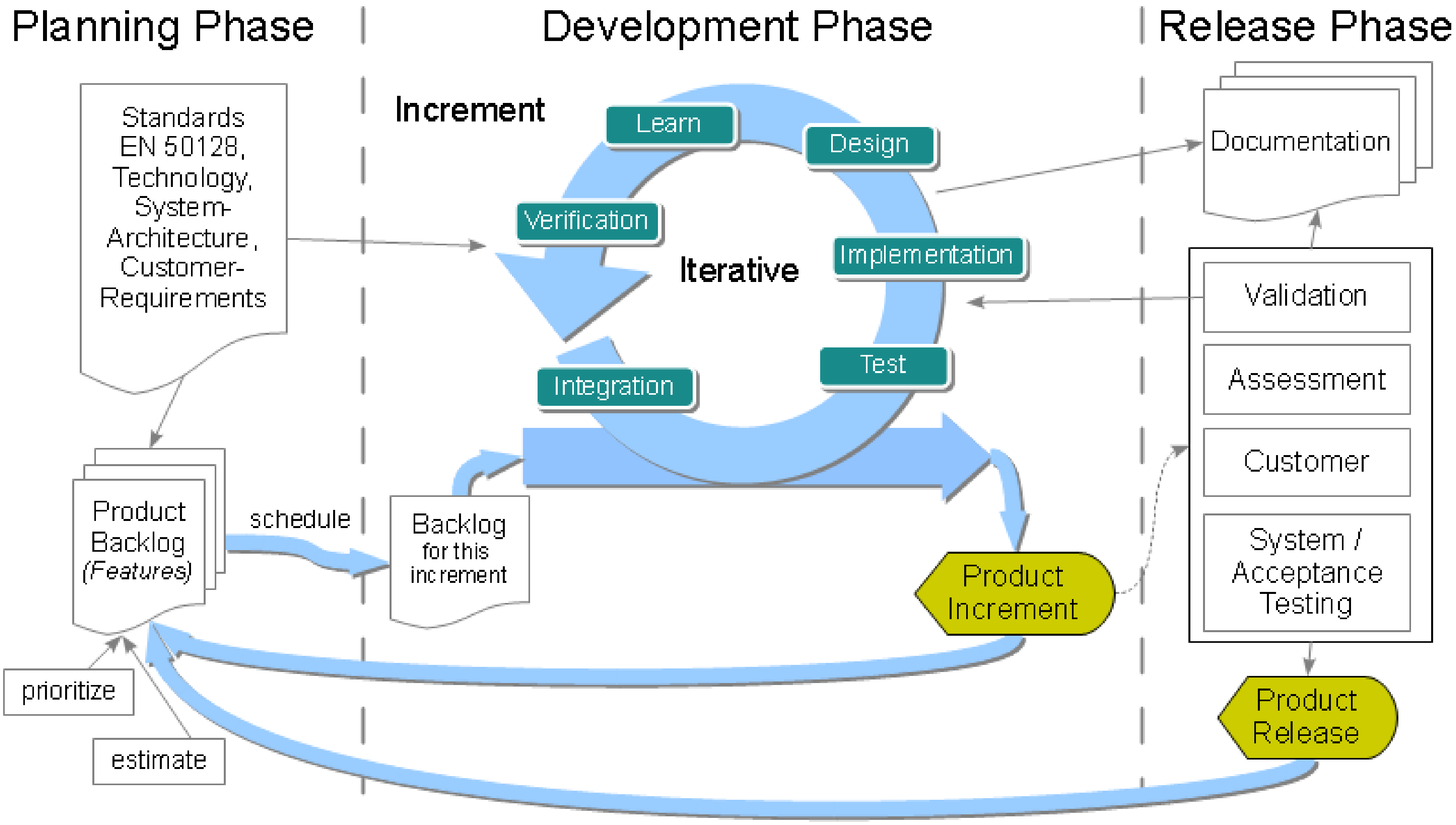
Results

- Standard and Agile – analogue objectives
  - Aim for quality
  - Big focus on testing

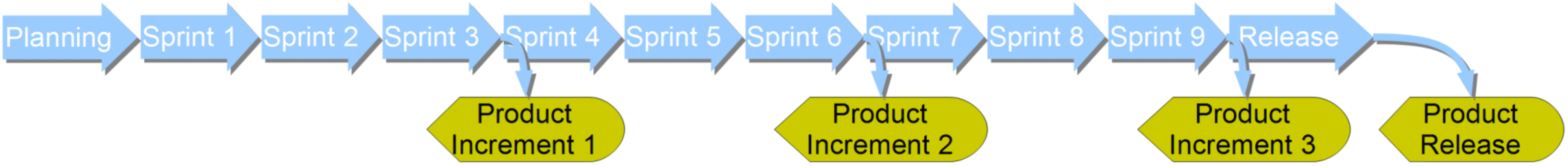
- Agile's incremental and iterative approach – allowed
- Simple agile approach needs enhancements
  - Additional assurance activities
  - More documentation
  - ✓ Agile methodologies are highly adaptable

➢ Define own process life cycle model

Safety-related agile software process model

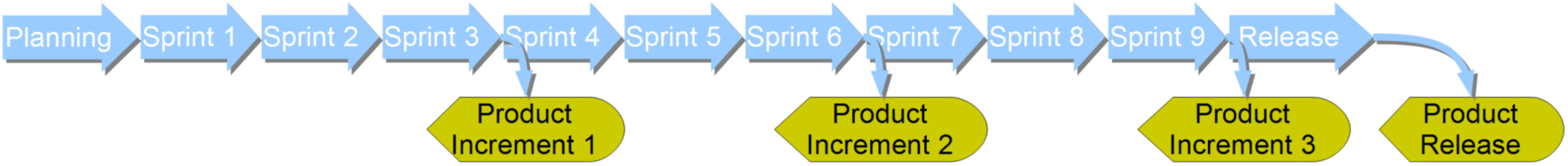


Development Life Cycle – Example 1



- **Planning Phase**
  - High level architecture
  - Safety concept
- **Sprint**
  - Incremental and iterative development
  - Definition of done
  - Documentation created during sprints
  - Safety spirit and verification

Development Life Cycle – Example 2



- **Product Increment**
  - System integration and testing
  - Customer presentations
  - Incremental validation
- **Product Release**
  - Final bug fixes
  - Assessment
  - Customer Acceptance Tests

Experiences from Thales

- **2010 First project - HMI development**
  - Scrum in development
  - Success – Quality improvement
    - Less bugs per development hour
    - “Stabilisation phase” shortened dramatically
    - Shortened bug fix period
  - Efficiency gain
    - High team spirit
    - Tester within the team
  - Still lots of documents
    - Work on automatic generation
  - Successful certification with CENELEC
- **2014 Scrum in all HMI development teams**

Challenges

- **Agile in large projects**
  - Synchronize on system level after each sprint
  - Mixture of traditional and agile teams
  - Distributed development
  - Advantage - early integration

- **Introducing in a traditional organisation**
  - More responsibility to the team
  - Less authority of project manager
  - Change of culture and mind-sets

Summary and Outlook

- **No disagreement – Use Agile Software Development**
  - In line with the safety standard
  - First successful projects experience
    - Improvements in cost, quality and time

- **Further introduction strategy for Thales**
  - Expand introduction to more development teams
  - Include system level
  - Involve customer closer
  - Learn from the experiences
  - Create customized trainings