Esri Developer Summit

March 26–29, 2012 | Palm Springs, California

esri.com/events/devsummit

Software Testing in the Agile World

Krishna Gummuluri

Agenda

- Process
  - What is Agile?
  - Agile SDLC
  - Traditional SDLC
  - How is Agile different?
  - Agile requirements strategy
    - Implications for testing
  - Agile testing strategies
  - Implications for testers
  - Agile quality strategies
- Test Engineering
  - Test design strategies
  - Test automation strategies

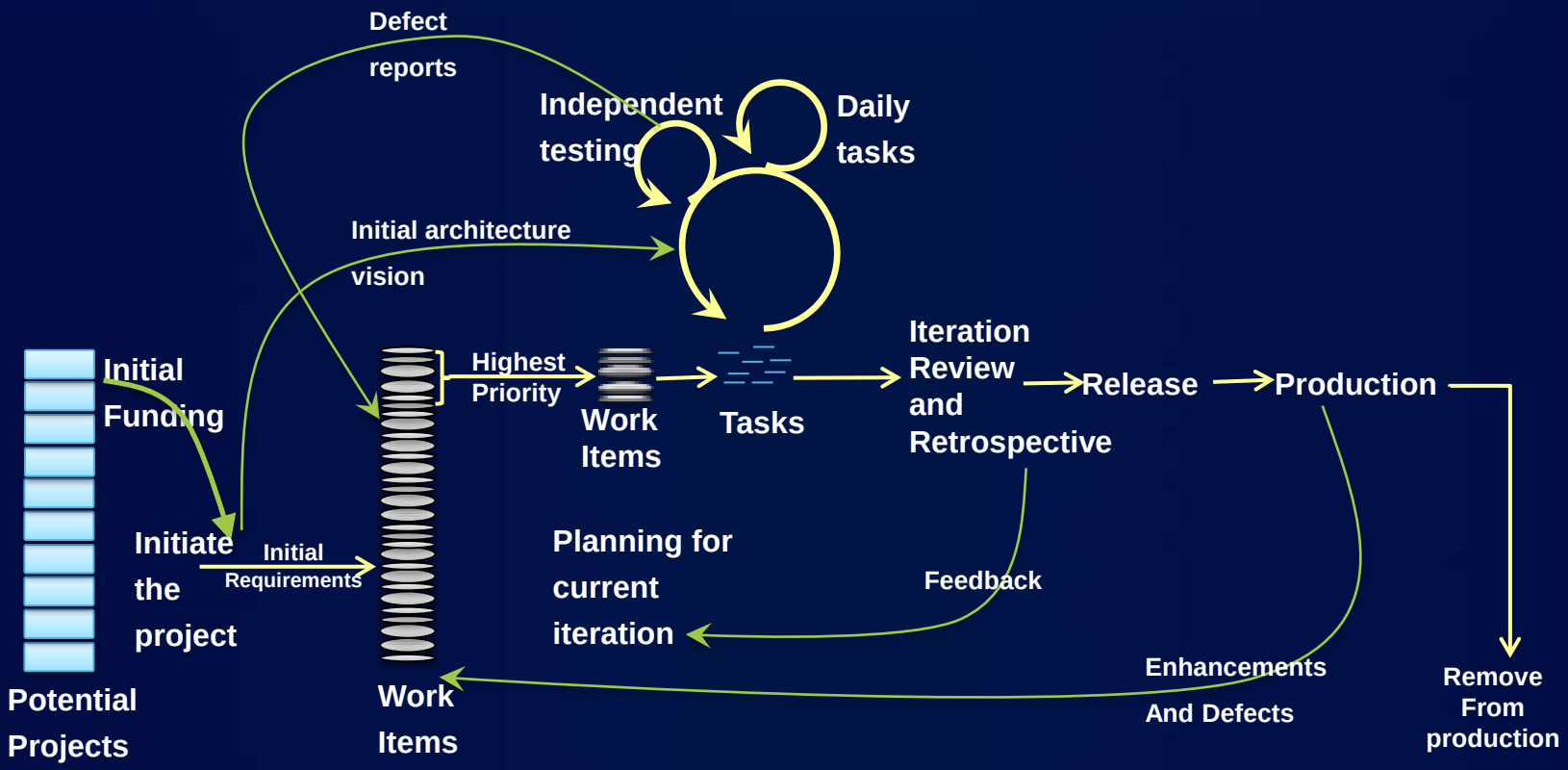
Agenda

- Process
  - What is Agile?
  - Agile SDLC
  - Traditional SDLC
  - How is Agile different?
  - Agile requirements strategy
    - Implications for testing
  - Agile testing strategies
  - Implications for testers
  - Agile quality strategies
- Test Engineering
  - Test design strategies
  - Test automation strategies

What is Agile?

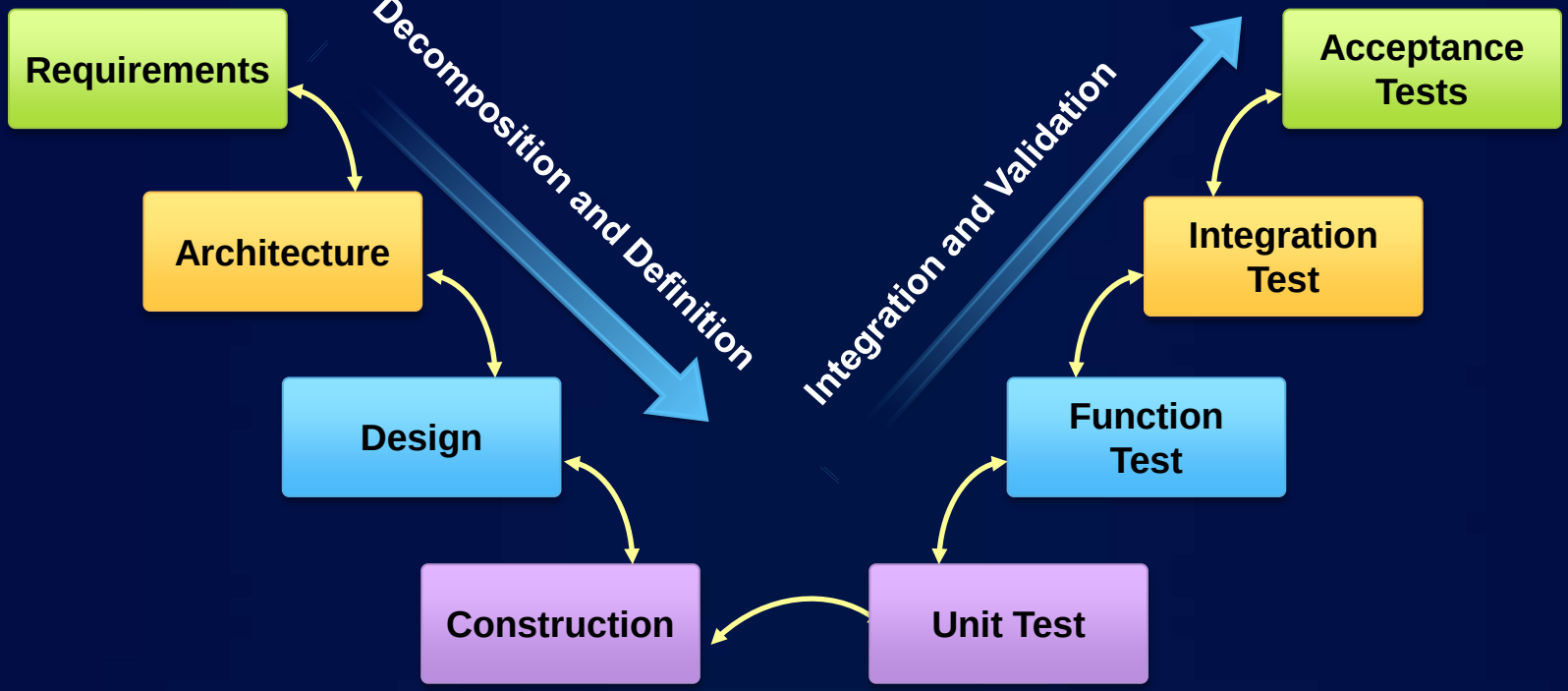
"An **iterative** and **incremental** (evolutionary) approach to software development which is performed in a highly collaborative manner by **self-organizing teams** within an effective governance framework, with just enough ceremony that produces high quality software in a cost effective and timely manner which meets the **changing needs** of its stakeholders."

Agile SDLC



Sprint -1	Sprint 0	Construction Sprints	Release	Production	Retirement

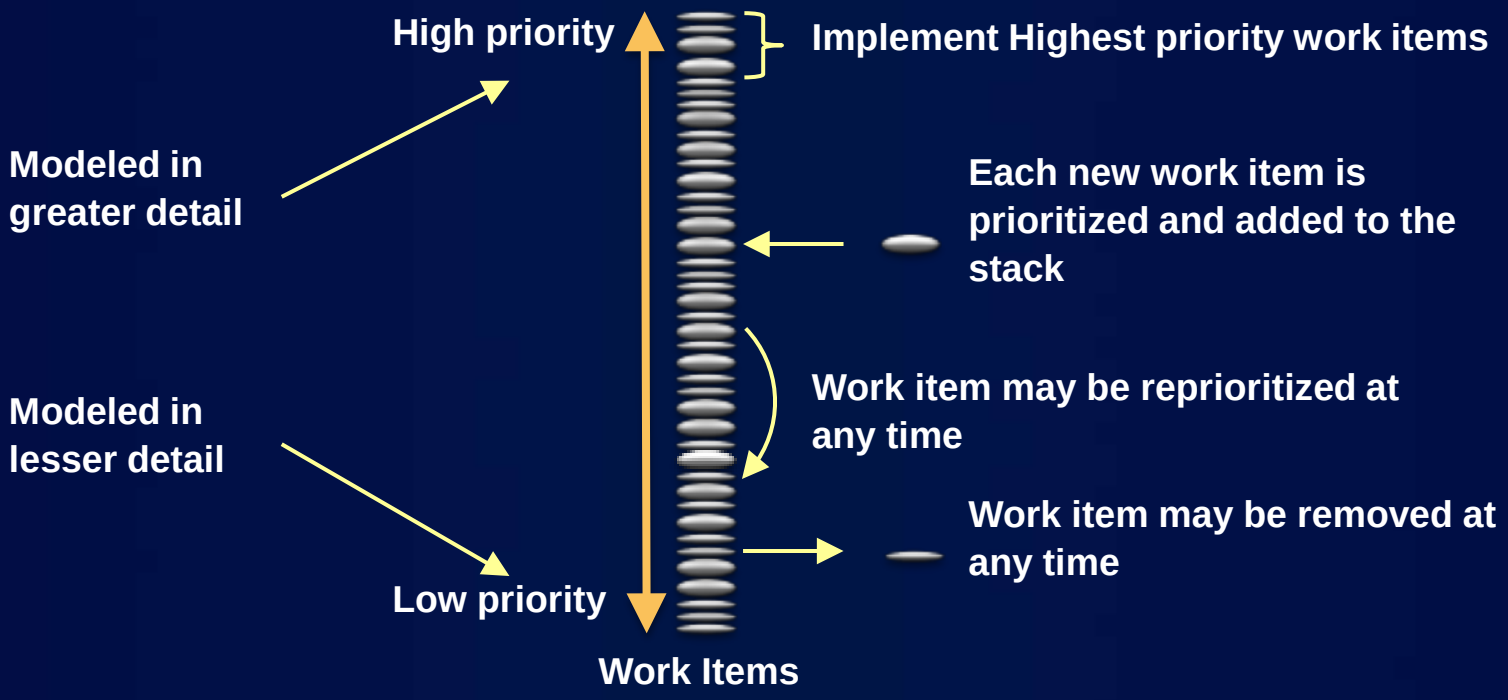
Traditional SDLC



How agile is different

- Greater Collaboration
- Shorter work cycle and constant feedback
- Need to embrace Change
- Greater flexibility
- Greater discipline
- Greater stakeholder accountability
- Greater range of skills

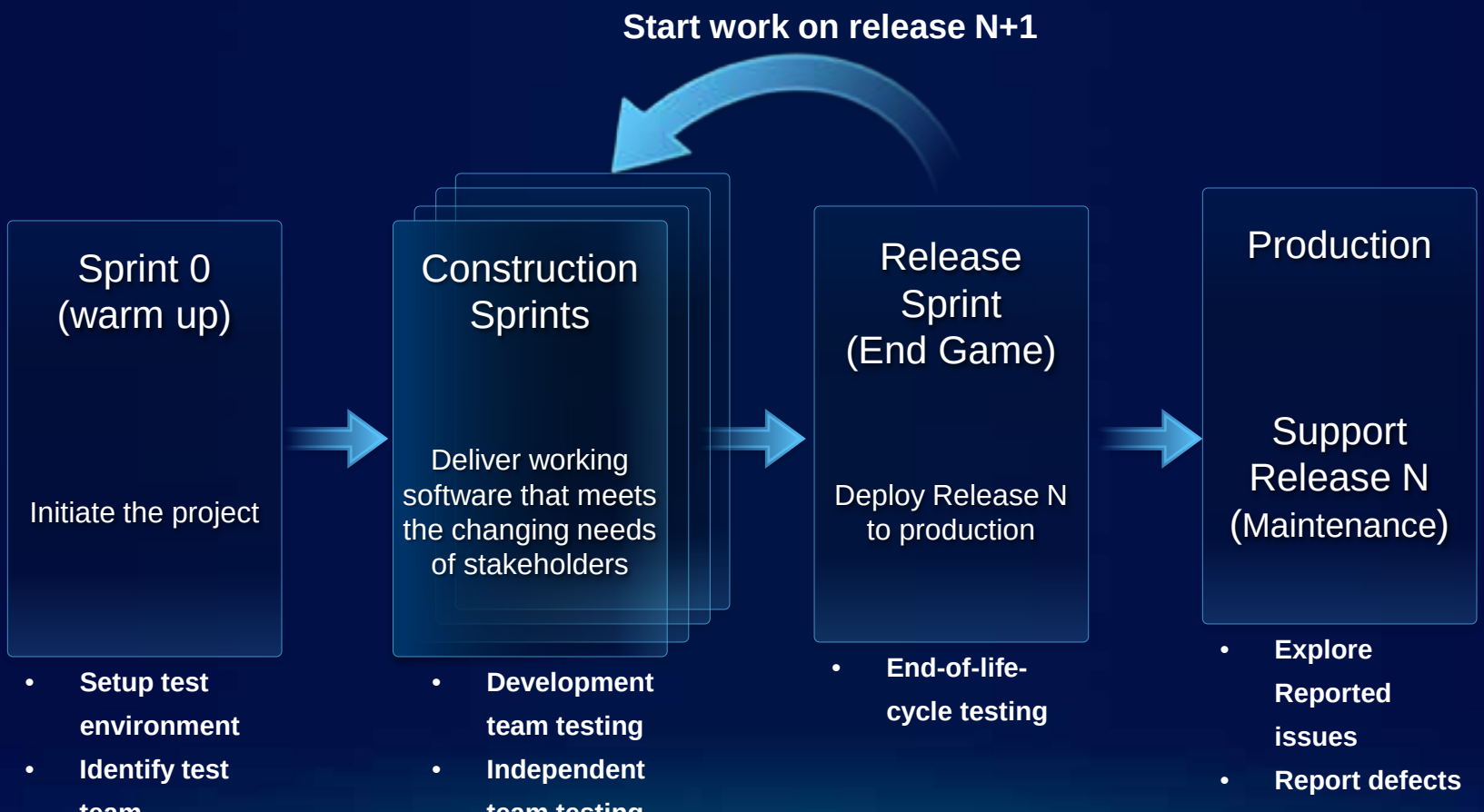
Agile requirements strategy



Implications for testing

- Agile testing must be iterative
- Testers cannot rely on having complete specification
- Agile testers must be flexible

Agile testing strategy - Testing throughout the SDLC



Set up test environment

- Test tools for developer testing
- Test tools for independent parallel testing
- Shared defect tracking system
- Hardware, Virtualization and lab management
- Continuous Integration tools

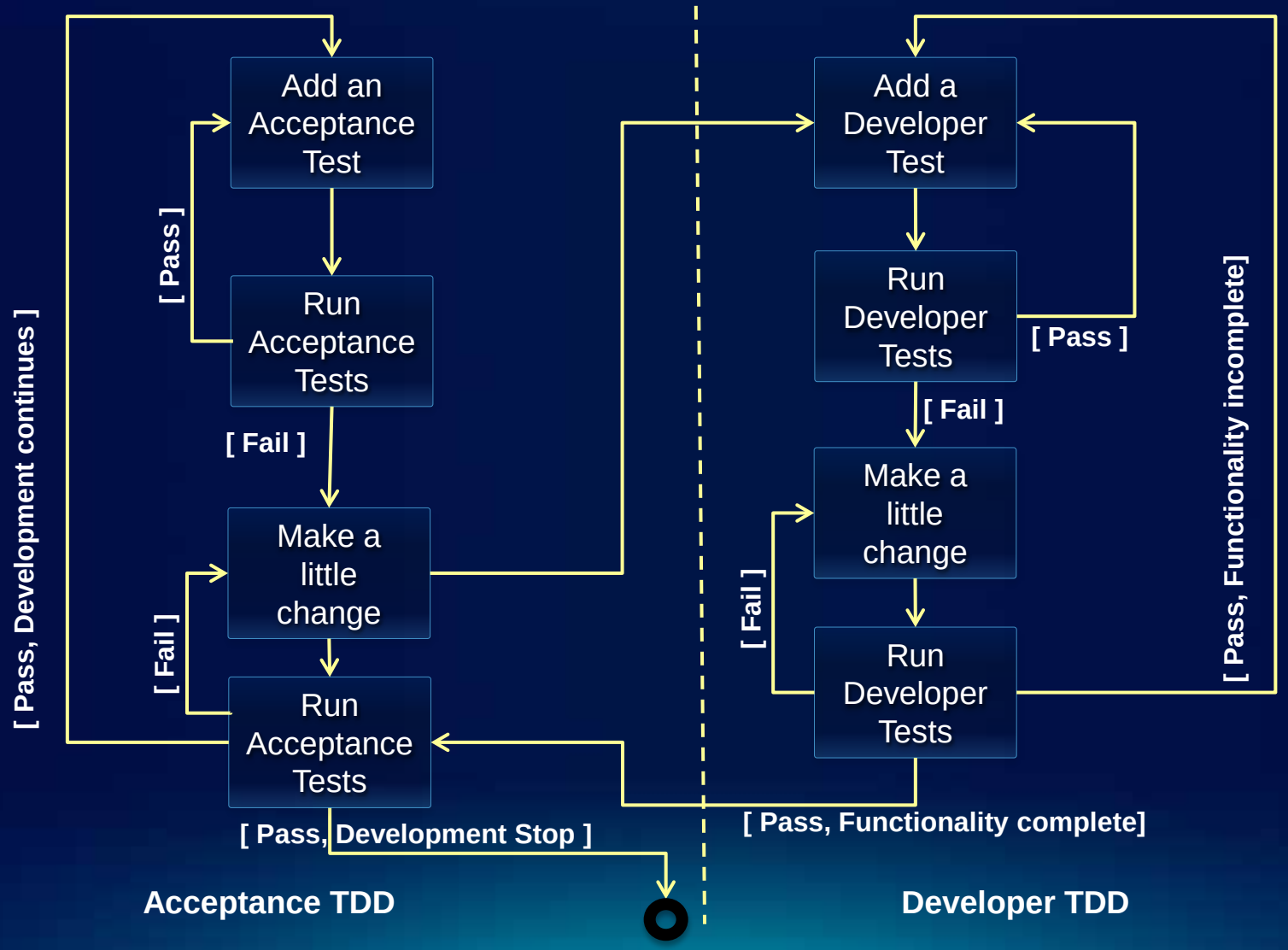
“Whole Team” approach

- Testers embedded in agile teams
- Multi-skilled team
- Flexible to contribute in any way then can
- Wider range of skills with one or more specialties
- Shorter feedback cycles
- “Sufficient” in straightforward situations
- Focus on “confirmatory” testing
- Issues
  - Group think
  - Lack of skills
  - Lack of knowledge of skills needed

“Whole Team” test strategies - Continuous Integration

- Build your system
- Run regression tests
- Perform static analysis
- Deploy “working” builds continuously

“Whole Team” test strategies - Test Driven Development

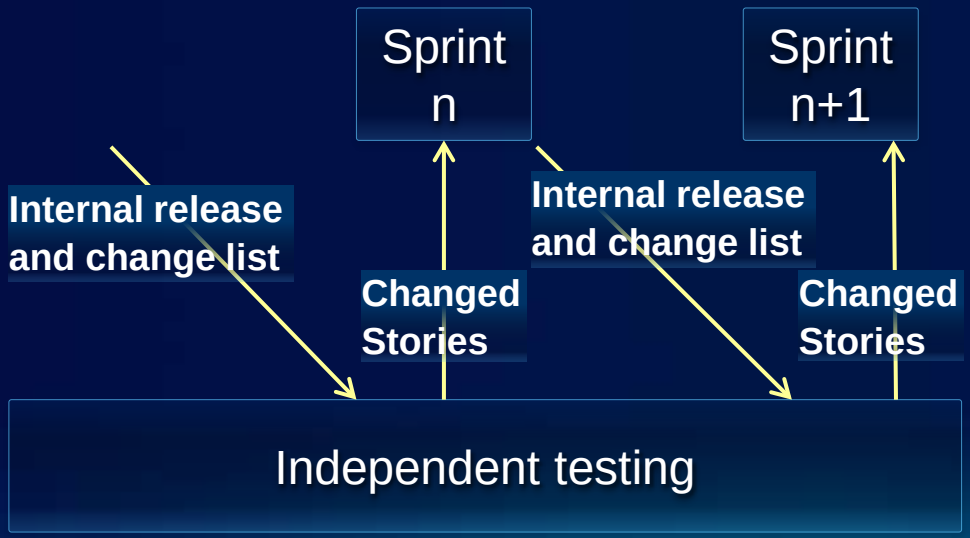


“Whole Team” test strategy - Test immediately after

- TDD requires discipline
- As good as TDD
  - If not days or weeks after.
  - Test coverage tools

Parallel independent testing team strategy

- Supplements “Whole Team” approach
- Complex environment
- Complex test scenarios
- Advanced forms of testing
- Support multiple development teams



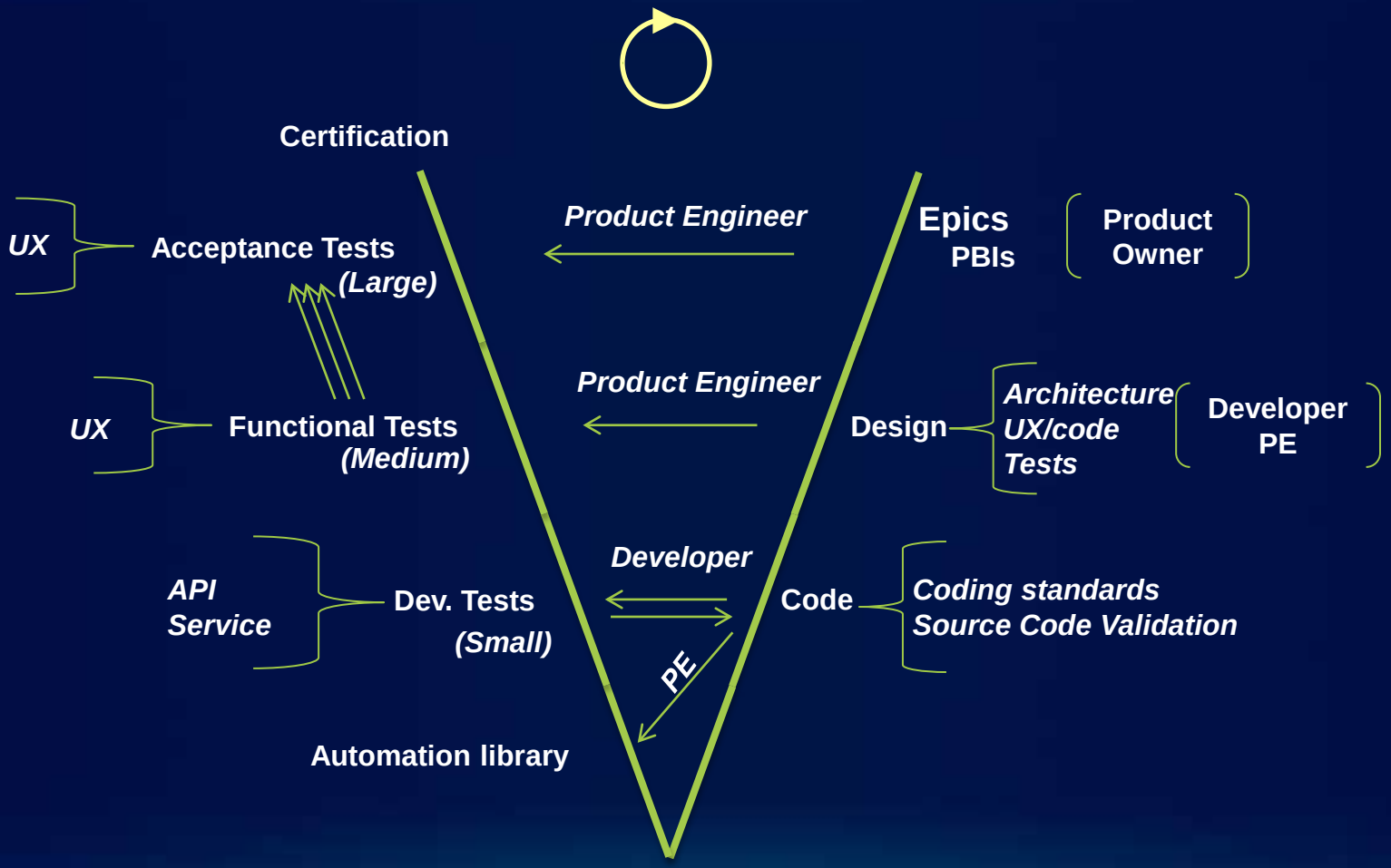
Confirmatory testing:

- Requirements TDD
- Design TDD

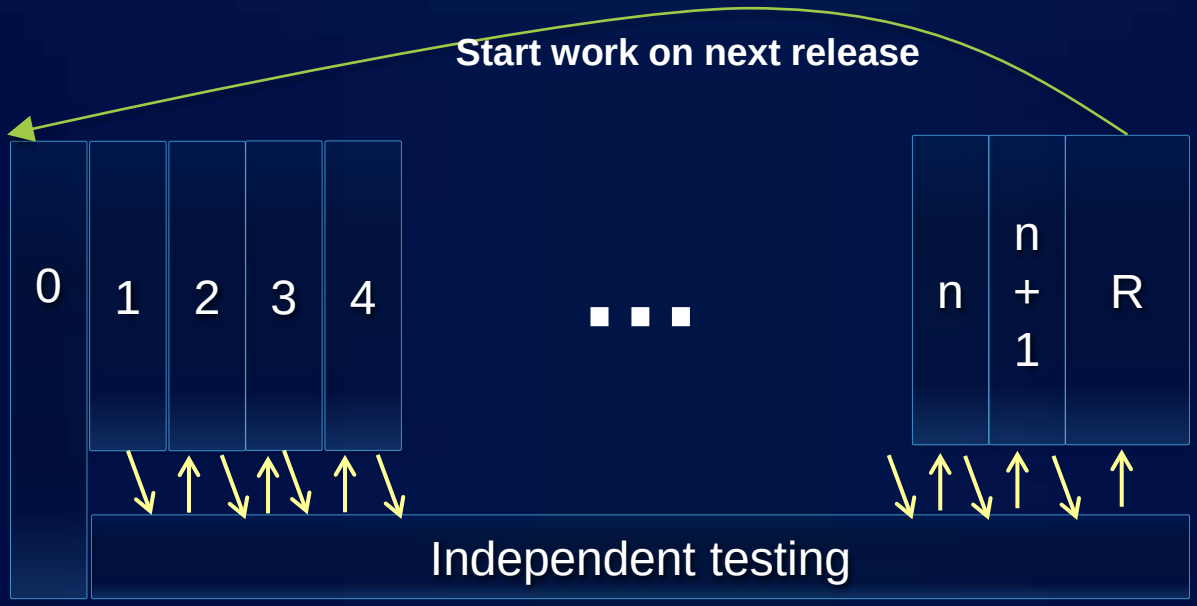
Independent testing:

- Acceptance testing
- Functional testing
- Exploratory testing
- Scenario testing
  - Documentation
  - Software
- System testing
- Usability testing

Testing strategy within construction sprints



End-of-life-cycle testing



- **Independent test team**
- **Very short**
- **Reasons**
  - **Professional**
  - **Legal obligation**
  - **Stakeholders requirement**
- **Required to scale agile**

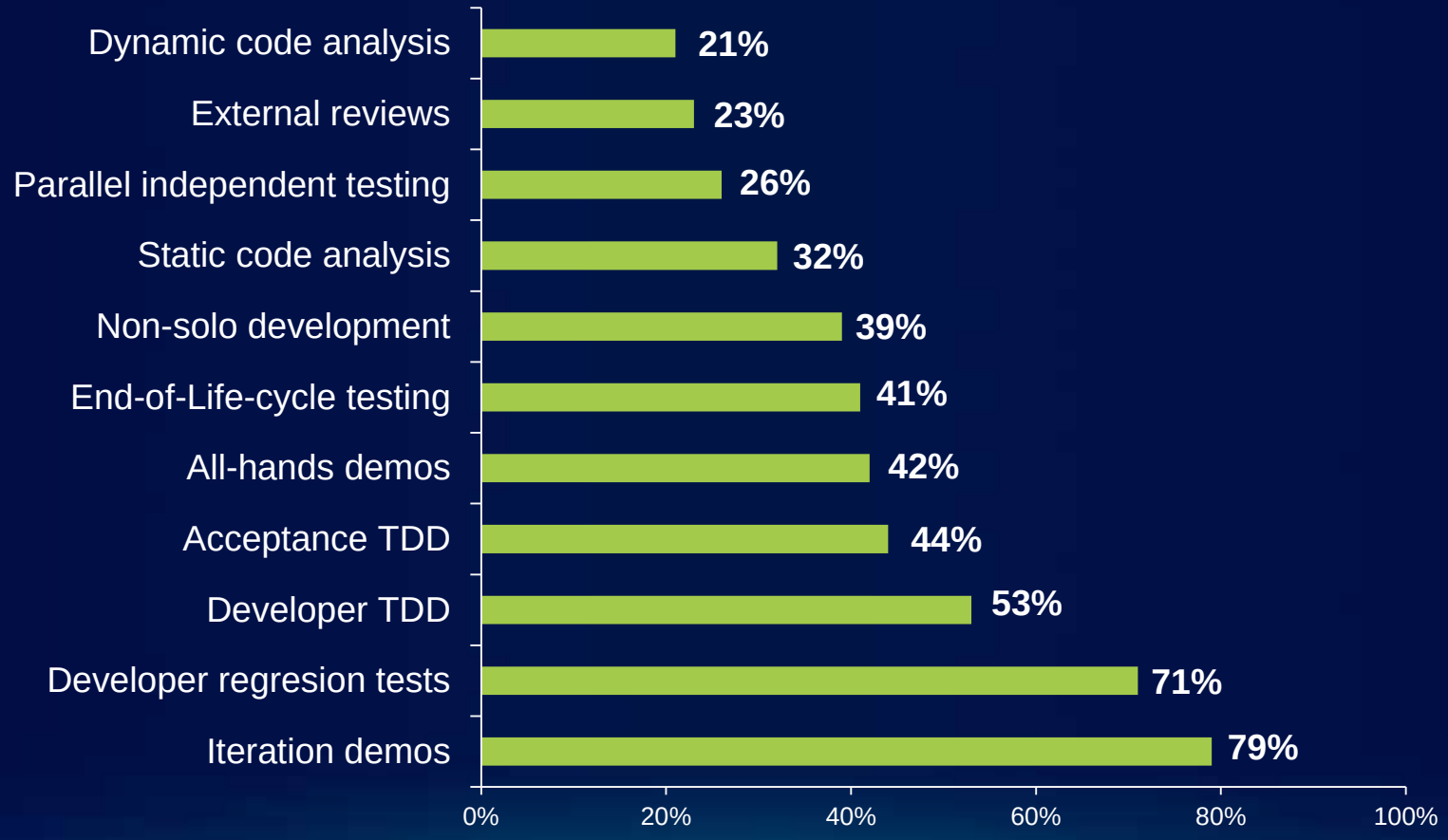
Implications for test practitioners

- Become generalizing specialists
- Be prepared to work closely with developers
- Focus on value added activities
- Be flexible

Agile quality strategies

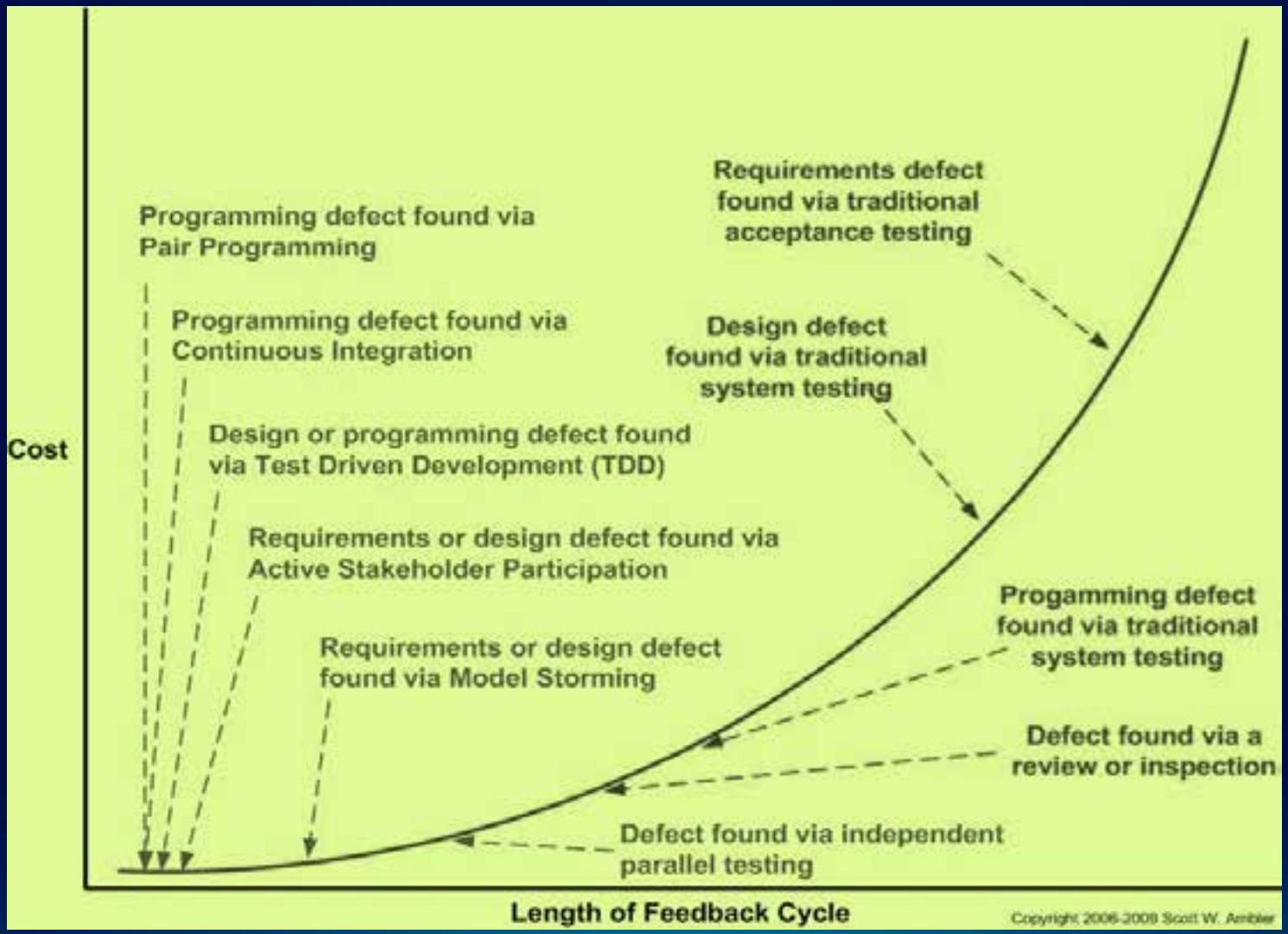
- Refactoring
- Non-solo development
- Static and dynamic code analysis
- Reviews and inspection
  - Iteration/sprint demos
  - All-hands demo
  - Light-weight milestone reviews
- Short feedback cycles
- Standards and guidelines

Testing practices amongst agile teams



Source 2010 How Agile Are You? Survey
www.ambysoft.com/surveys

Why Agile strategies work



Agenda

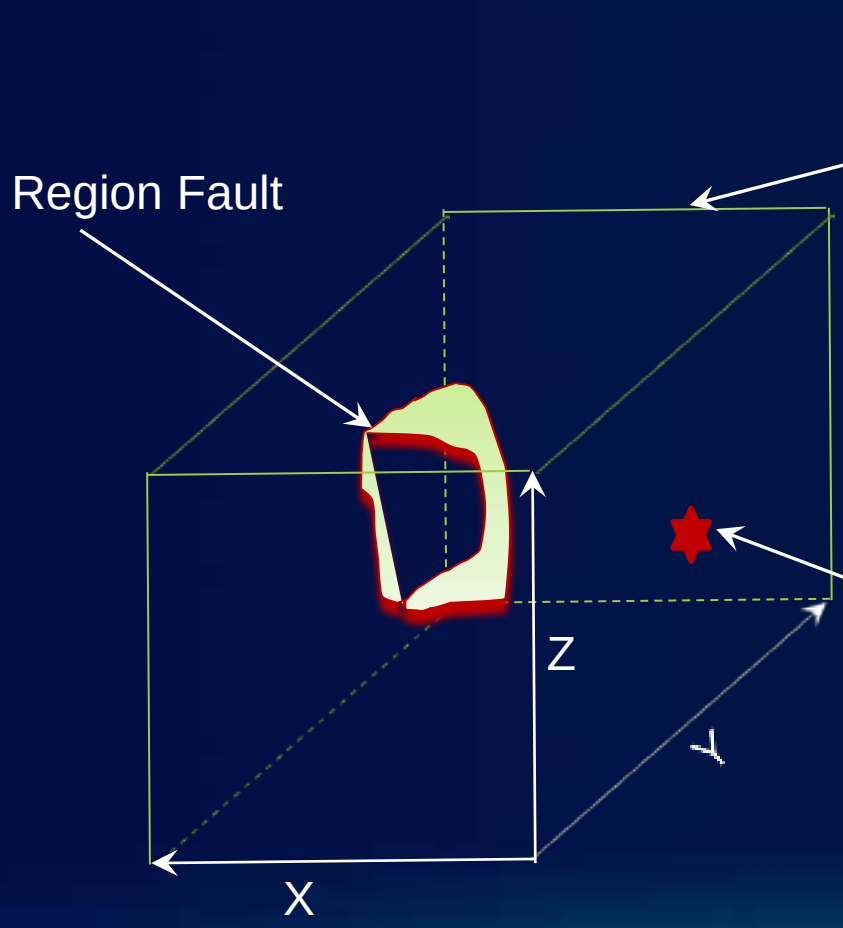
- Process
  - What is Agile?
  - Agile SDLC
  - Traditional SDLC
  - How is Agile different?
  - Agile requirements strategy
    - Implications for testing
  - Agile testing strategies
  - Implications for testers
  - Agile quality strategies
- Test Engineering
  - Test design strategies
  - Test automation strategies

Test design strategies – Acceptance Criteria categories

- Functional
- Integration/Touch points
- Accessibility
- L10N/I18N
- Performance
- Scalability
- Security
- Platform Awareness
- Help Documentation

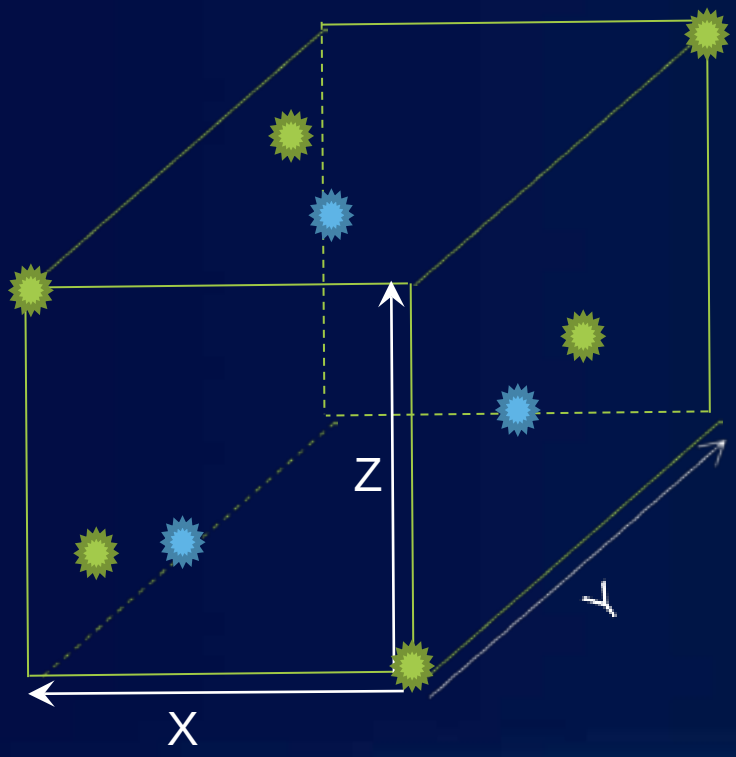
Test design strategies – Functional tests

- Equivalence classes
- Boundary value analysis
- Decision tables
- State transition
- Pairwise testing
  - http://msdn.microsoft.com/enus/library/cc150619.aspx
  - http://aetgweb.argreenhouse.com/

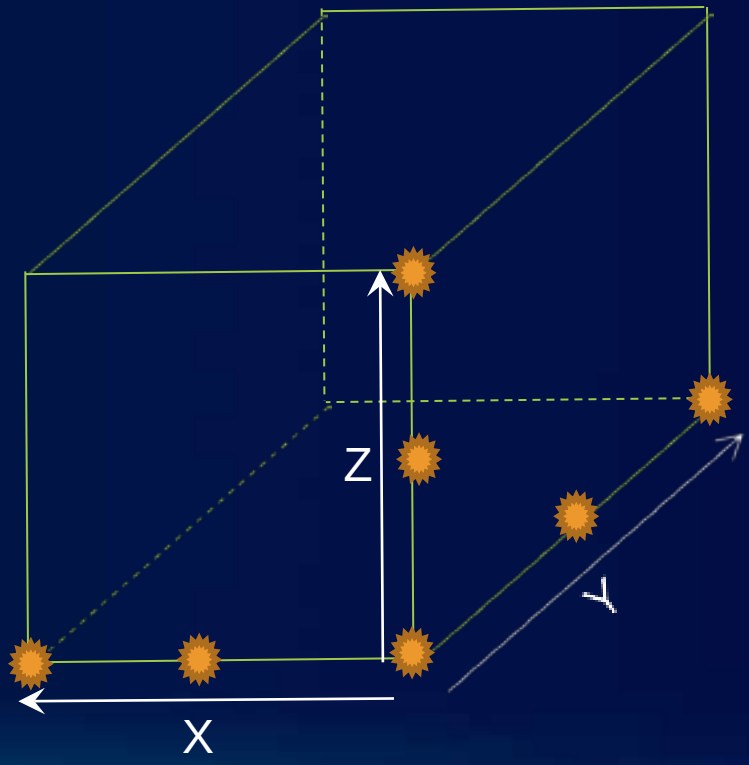


X	Y	X
X1	Y1	Z1
X2	Y2	Z2
X3	Y3	Z3

Functional coverage and defect detection potential



Pair wise or OA based testing



One factor at a time testing

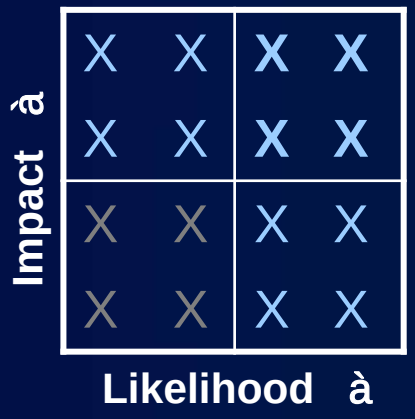
Test Design Demo

Test automation strategy

- Adopt a planned approach to developing test automation
- Increase the quality of test automation code
- Promote code reuse across teams by increasing the awareness of test automation code that is available for reuse. For example, by document test library methods in code and publishing to a wiki
- Decrease maintenance cost
- Increase test automation coverage beyond functional testing (e.g. performance and stability testing or localization testing)

Selecting test cases for automation

- **Risk Based**



- **Incidental execution of the test area**
- **How long it takes to run the test manually**
- **What is the cost of automating the test**
- **How easy is the test cases to automate**
- **How many times is the test expected to run in a project**

Test Automation Demo

Please fill out survey forms!
Thank you

Any Questions?