Levers for changing software delivery

v1.0.4

Michiel Kalkman

SDLC

SDLC

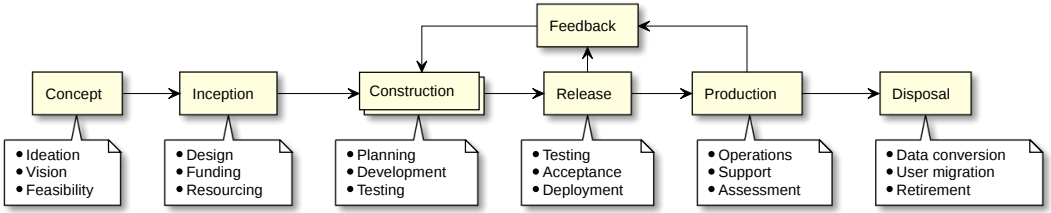

Figure 1: Phases of the System Development Lifecycle

General Observations

- Phases are ordered sequentially
  - With feedback loops to register defects
- Phases have distinct functions
  - Transitions between phases require a context switch
- Work in earlier phases affects later phases
  - Distance between defect creation and detection is doubled via feedback

General Goals

We want to,

- Minimise feedback volume, rate and processing time
- Minimise the cost of phase transitions
- Minimise the distance between defect creation and detection
- Minimise the volume of work passing through the cycle

SDLC improvements

Project Phases

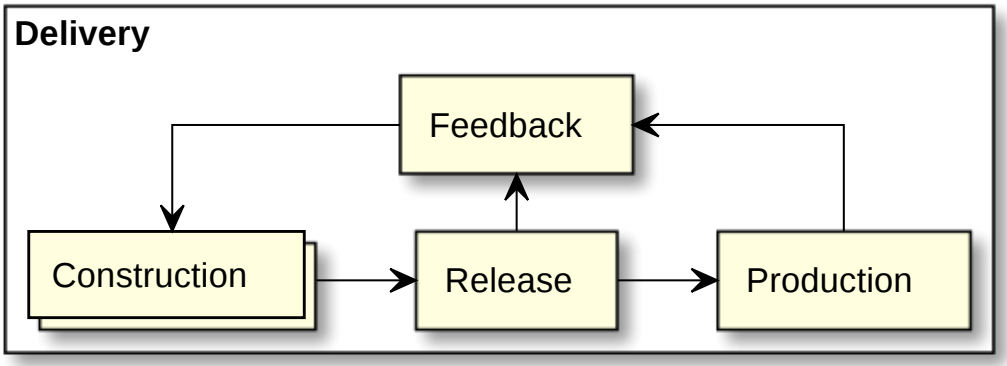

Figure 2: The DevOps Cycle

Control : Add Quality Gates

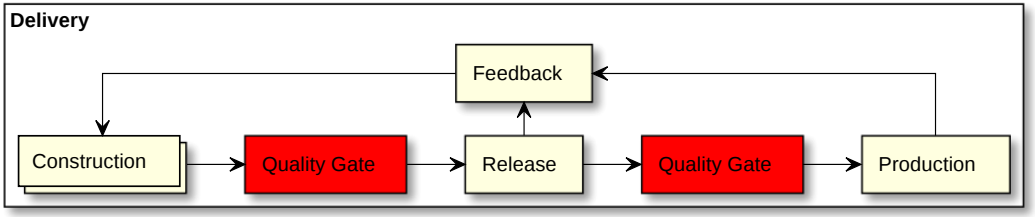

Figure 3: Add quality gates

- **Cost**
  - Raise the cost of phase transitions
- **Benefits**
  - Lower feedback volume and rate
  - Lower the average distance between defect creation and detection

Control : Improve Defect traceability

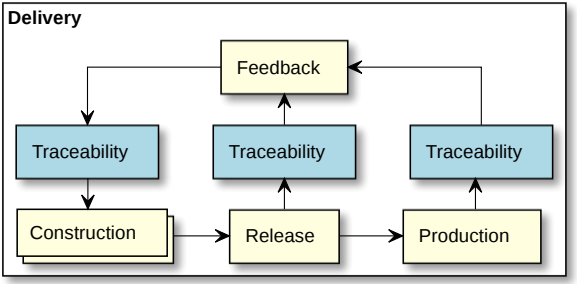

Figure 4: Tracing

- **Benefits**
  - Lower feedback processing time - fast remediation
- **No effect**
  - Cost of phase transitions, feedback volume, defect creation/detection distance

Option : Improve Transition Throughput

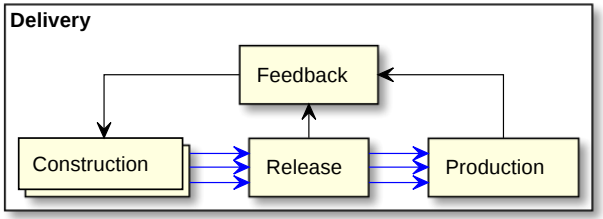

Figure 5: Parallelisation

- **Benefits**
  - Cost of phase transitions, feedback time, defect creation/detection distance
- **No effect**
  - Feedback, defect creation/detection distance, volume of work

Option : Improve Phase Output

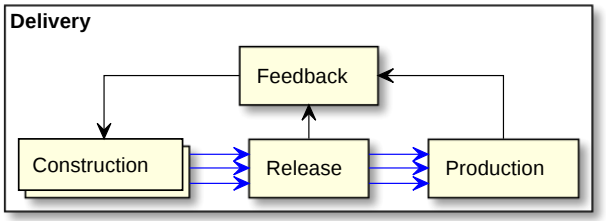

Figure 6: Tracing

- **Benefits**
  - Minimise feedback volume
  - Minimise the volume of work passing through the cycle
- **No effect**
  - Lower cost of phase transitions

Combined model

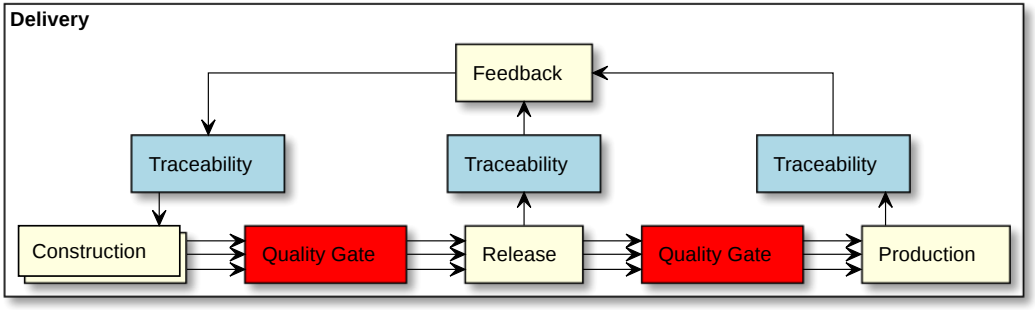

Figure 7: Combined

Phase improvements

Classic Testing V-Model

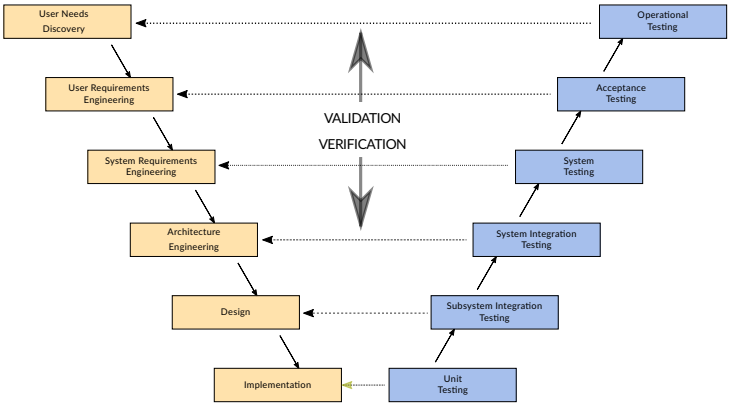

Figure 8: Testing V-Model - Full waterfall project

Testing V-Model - Single Feature

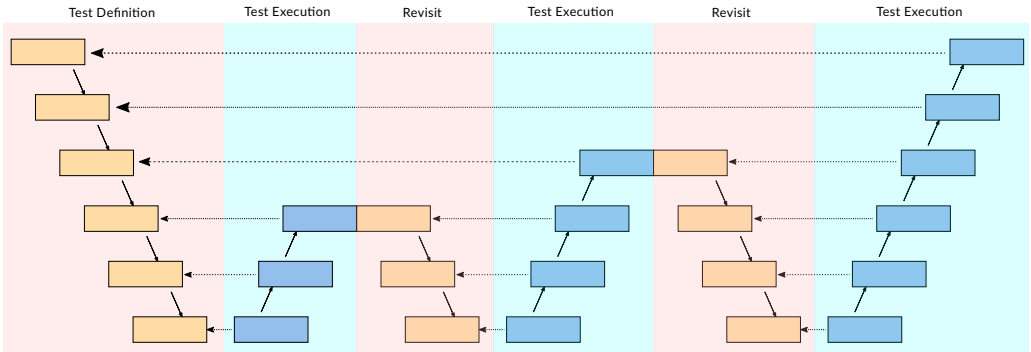

Figure 9: Testing overlay - separate test definition and execution, fail fast and iterate

Testing - Independent Feature Development

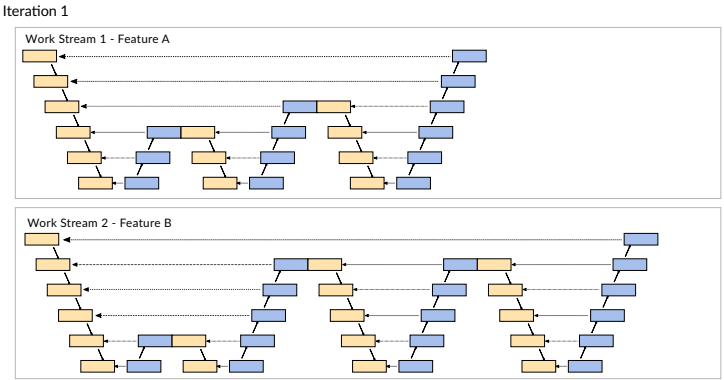

Figure 10: Testing V-Model

Testing - Multiple Development Streams

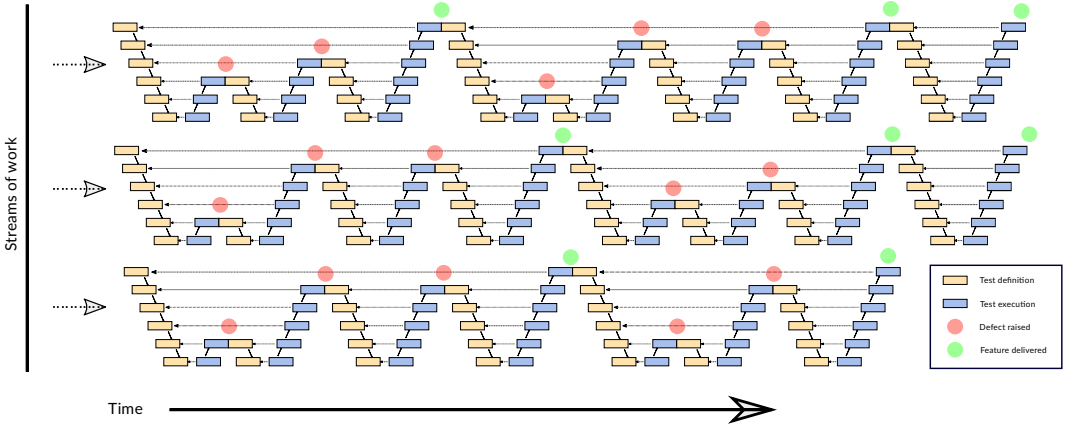

Figure 11: Left-Shifted Testing - Find Defects Early

Shift Left Testing

Goal

- Reduce the average distance between defect creation and detection
- Separate test definition from test execution

What	How
Define test requirements at each step	Engage test team at each step
Execute tests as soon as possible	Gate progress at each step on testing
Minimise the test execution barriers	Automate testing, traceability, reporting
Minimise the test execution time	Automate testing, make test runs cheap
Have test execution consistency	Automate testing, make test runs predictable

Challenges

Technical Debt

Causes

- Team ramp up, learning curve - Immature engineering practices
- Time-to-market drive - Delivery pressure shortcuts
- Progressive learnings - Mismatch between technology and application
- Product repositioning - Mismatch between platform and application

Effects

- Lost opportunity, wasted effort
- Testing overhead - Complexity doesn't increase linearly
- Development overhead - Any change is more complex than necessary
- Still has to be maintained - Any maintenance is more complex than necessary
  - Industry average 15 to 50 errors per 1000 **delivered** LoC
    - *From Code Complete, Microsoft Press*
- Compound interest on the debt

Pressure of re-work and technical debt

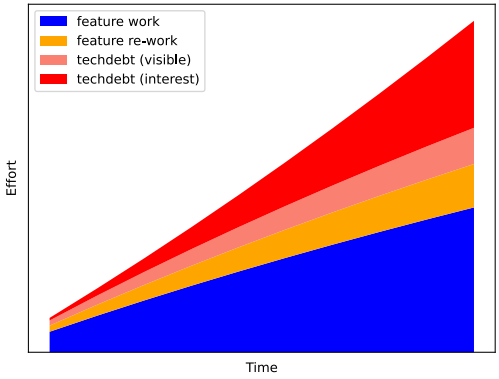

Figure 12: Cumulative flow of re-work and technical debt pressures

- Re-work is lost opportunity
- Interest on technical debt compounds
- Track both new debt and its interest pressure
- Track re-work and cycle times

Pre-project feedback

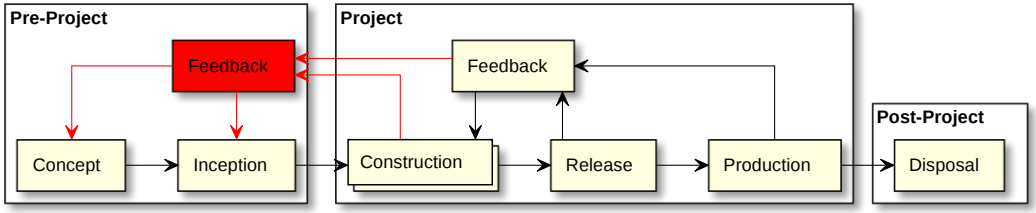

Figure 13: Concept feedback

- No well-defined feedback mechanisms
- No clear outcomes
- Politics
- Move more decisions to Construction phase pre-project

Non-feature budget expenditure

Process changes require significant expenditure of resources. Find secondary benefits that address pain points or have a financial impact to lower the cost.

- Improving Traceability for Defects can also be used to lower to cost of Support
- Automation and parallelisation can be used to scale Development

ROI of Manual vs Automated Testing

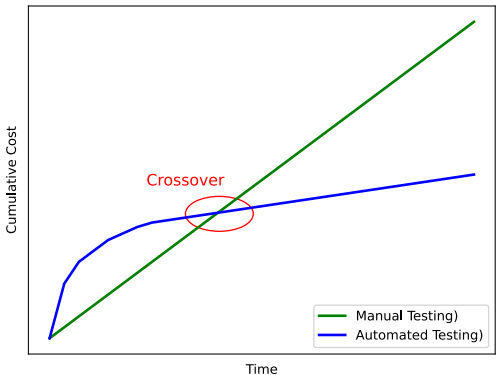

Figure 14: By stage

- There is always an upfront investment with automation
- Understand where the crossover with manual testing is
- Make this visible to stakeholders
- Keep track of and report on the value of automation as the project progresses

Value of work by stage

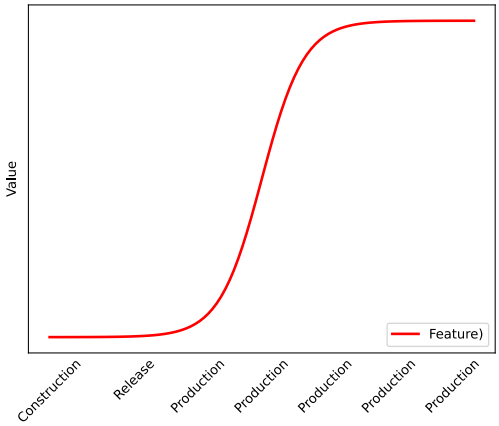

Figure 15: By stage

- Get completed features to production as soon as possible, **no excuses**
- Use feature flags to gate activation if necessary

Summary

Summary

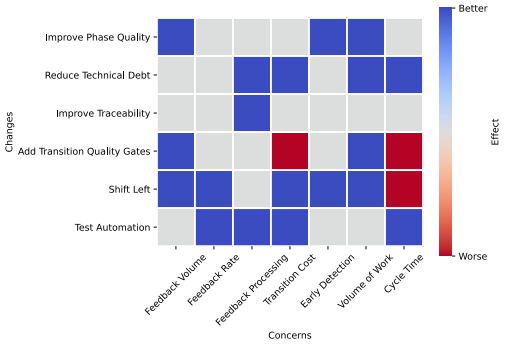

Figure 16: Effect of changes across concerns

- Understand which dimension needs to change
- Assess change proposals accordingly
- Can't change everything at once
- Focus on outcome

Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)