Building Reusable Testing Assets for a Product Line

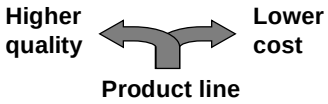
John D. McGregor
Visiting Scientist - SEI
Senior Partner - Korson-McGregor
Associate Professor - Clemson University
johnmc@cs.clemson.edu

Qualifications

- Worked on large software systems - 500 developer FTE
- Worked on early product line efforts
- Working on current product line efforts
- Domains:
  - telephony
  - wireless
  - insurance
  - ...

Motivation

- Effort estimates range from 50 - 200% that of the actual production system
- If reuse of test assets can reduce this by 10% this represents substantial savings
- In fact we can reduce the effort by considerably more than that depending upon the initial maturity



Outline

- Overview
- Definitions
- Activities & Assets
- An example company

Product Line Definition

- A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. (From A Framework for Software Product Line Practice, v3; SEI, ©2000)
  - Independent product cycles
  - Core asset base

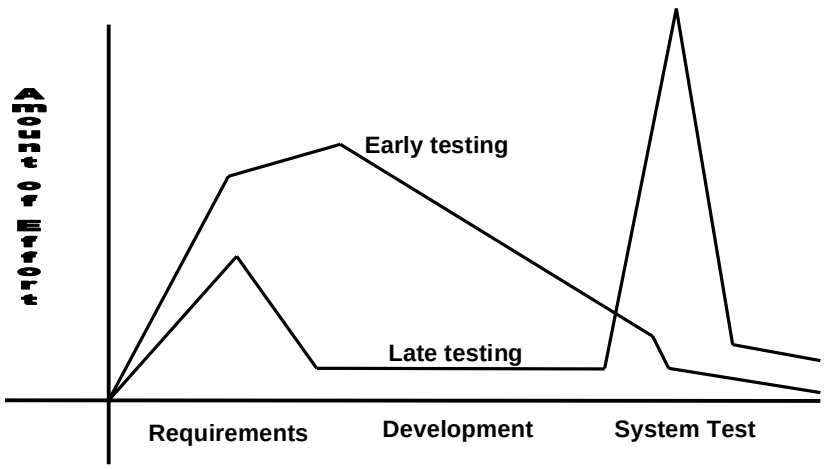
Test Assets

- The basic test assets are
  - test plans
    - summary of strategies for test case selection
  - test cases
    - <pre-conditions, input stimulus, expected results>
  - test reports
    - standard summary of results of the tests that have been run

Reusable Assets

- Making assets reusable requires both management and technical support
  - resources must be allocated on a different schedule which changes management practice
  - "correct" design choices are different
- Reuse can occur both vertically between the product line and the products and horizontally between products
  - vertical reuse propagates interfaces
  - horizontal reuse propagates opportunistic components

Early Testing



Comprehensive Testing

- Testing is conducted during and after development
  - during development we are concerned that the work does what it does correctly
  - after development we are concerned that it does what it is supposed to do
- Testing is conducted before and after code is written
  - before code is written test cases guide the inspection technique
  - after code is written test cases are executed using the product code

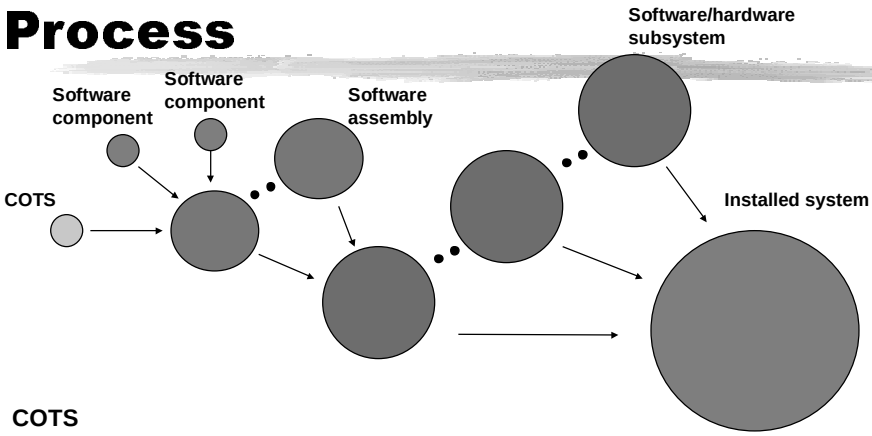
Outline

- Overview
- Definitions
- Activities & Assets
- An example company

Definitions

- Testing - the detailed examination of a product guided by specific information
- Tester - any person conducting a test
- Developer - the person constructing the product under test
- Domain expert - person knowledgeable in the subject matter of the application

Basic Testing Process



Test Points - 1

- Requirements model
- Analysis and design models
  - Domain analysis
  - Application analysis
  - Architectural design
  - Mechanism designs
  - Detailed designs
- Individual components
  - Interfaces
  - Implementations

Test Points - 2

- Clusters of components
  - interactions
- Increments
  - end user functionality
- Complete developed systems
  - comparison to requirements model
- Deployed systems
  - interaction with other installed applications

Exercise

- Think about your own development process.
  - Make a list of the major steps.
  - Identify the testing activities for each development step.
  - What types of defects could be escaping from these steps?

Development Activity	Testing Activity

Reusable Test Assets

- Knowledge
  - what can go wrong
  - techniques
  - patterns
- Models
  - document templates
  - specifications that appear multiple times
- Code
  - test case implementations

IEEE Test Plan - 1

- **Introduction**
- **Test Items**
- **Tested Features**
- **Features Not Tested (per cycle)**
- **Testing Strategy and Approach**
  - Syntax
  - Description of Functionality
  - Arguments for tests
  - Expected Output
  - Specific Exclusions
  - Dependencies
  - Test Case Success/Failure Criteria

IEEE Test Plan - 2

- **Pass/Fail Criteria for the Complete Test Cycle**
- **Entrance Criteria/Exit Criteria**
- **Test Suspension Criteria and Resumption Requirements**
- **Test Deliverables/Status Communications Vehicles**
- **Testing Tasks**
- **Hardware and Software Requirements**
- **Problem Determination and Correction Responsibilities**
- **Staffing and Training Needs/Assignments**
- **Test Schedules**
- **Risks and Contingencies**
- **Approvals**

Test Case

- <pre-conditions, input stimulus, expected results>
- pre-conditions
  - need 35,000 transactions to be realistic
- input stimulus
  - run the "balance sheet" report
- expected results
  - must independently verify what the result should be
  - trace through 35,000 transactions
- reuse of this test case will be VERY cost effective

Test Report

- Test reports are more important in some domains than others
- Serve as legal proof of meeting obligations
- Present results
  - Individual results
  - Summarized results
- Analysis of test effectiveness

Other Testing Definitions

- Test technique
  - technical processes for test activities
  - example - combinatorial test design
- Test coverage
  - a measure of how much of the test item has been examined by the tests
  - example - one test per use case

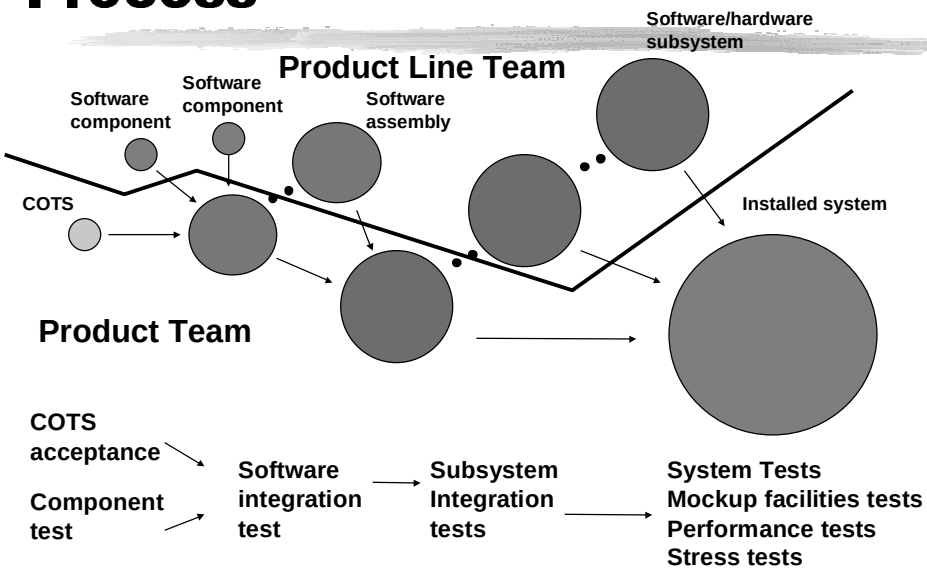
Exercise

- Create a list of the test documents that your current project creates
- Are there documents that you produce that have not been discussed? Describe their usefulness.
- Are there documents that we have discussed that your project does not use? Is the information captured elsewhere? What might you be missing because of not having this document?

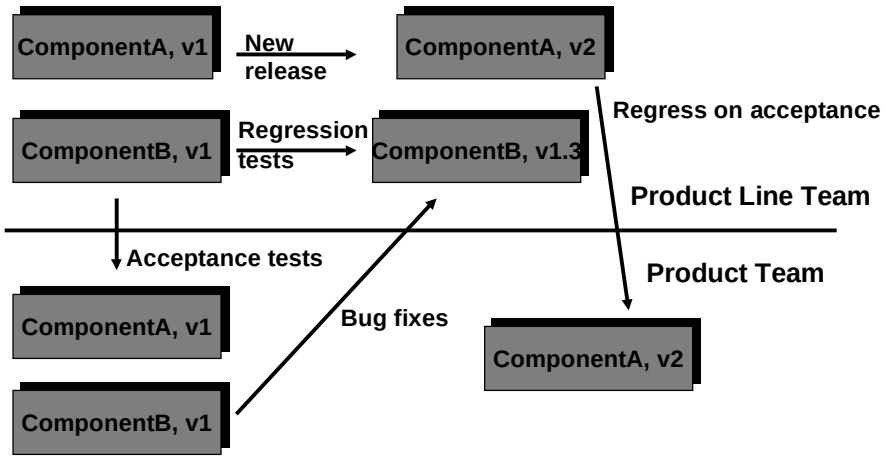
Outline

- Overview
- Definitions
- Activities & Assets
- An example company

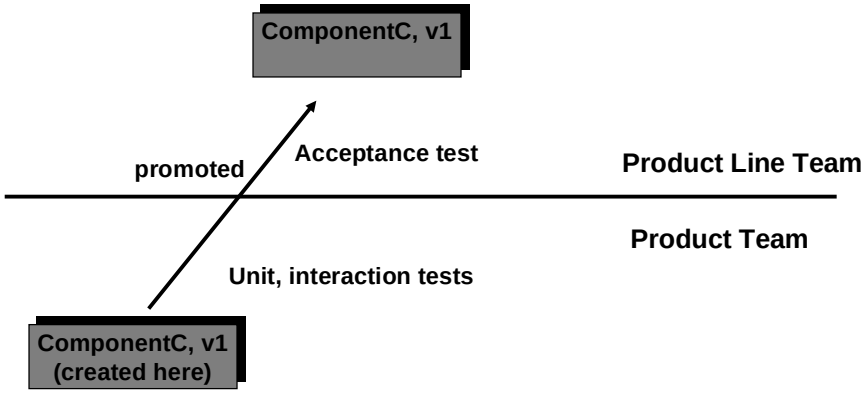
Basic Product Line Testing Process



New Test Points



New Test Points



Basic Reuse Principles

- Encapsulation
  - Package for portability
- Information Hiding
  - Separation of what from how
- Abstraction
  - Remove detail to focus on essentials

These are applied at each organizational level.

Activities by Company Levels

- Organizational Management
  - test strategies
  - test standards
- Technical Management
  - test process
  - assign roles
  - construct test plan
- Software Engineering
  - construct test infrastructure
  - execute test plan

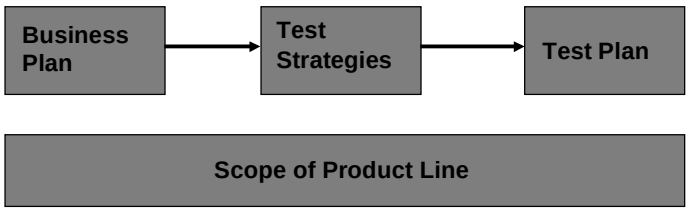
Organizational Management Activities

- Strategic level
  - organize for reuse of assets
  - integrate development and testing
- Set company-level standards
  - test coverage
  - estimated remaining defects

Organizational Management Activities

- Encapsulation
  - business plan provides a single basis for strategies
- Information Hiding
  - strategies define what is to be achieved not how to achieve it
  - reliability, safety, ...
- Abstraction
  - separates test plans and strategies into levels of responsibilities

Organizational Management Activities



Exercise

- What test strategies are used in your domain currently?
  - Levels of coverage that are required
  - Method of sampling
- How might these be affected by a product line approach?

Technical Management Activities

- Establish processes
  - test process for each level
- Assign roles in testing processes
  - define roles and associate w/activities
- Construct test plan
  - test activity for each development step

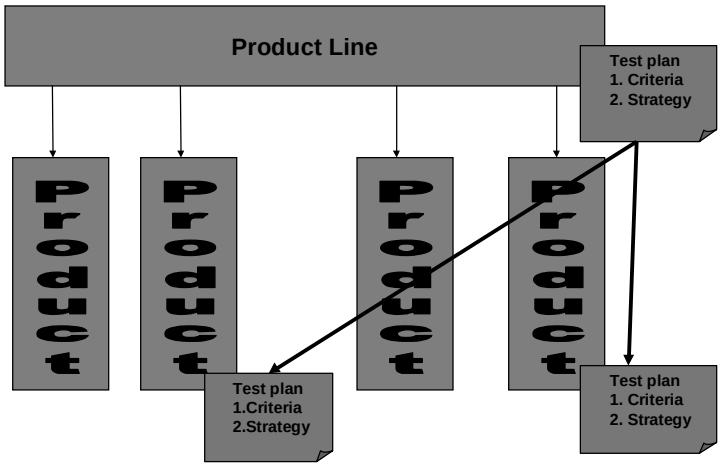
Technical Management Activities

- Encapsulation
  - System test plans are created on a per use case basis
  - all required information is bundled together
- Information Hiding
  - Test plans are specified but not implemented
- Abstraction
  - The test assets are defined in levels of abstraction
  - The levels correspond to the levels in the design of the software

Creating the Test Plan

- The plan describes how test assets will be propagated from the product line to the product teams
- The plan assigns responsibilities to personnel for creation and maintenance

Test Plan Reuse



Test Plan Reuse

- **Testing Strategy and Approach**
  - Syntax
  - Description of Functionality
  - Arguments for tests
  - Expected Output
  - Specific Exclusions
  - Dependencies
  - Test Case Success/Failure Criteria
- **Pass/Fail Criteria for the Complete Test Cycle**
- **Test Suspension Criteria and Resumption Requirements**
- **Testing Tasks**
- **Hardware and Software Requirements**
- **Staffing and Training Needs/Assignments**
- **Test Schedules**
- **Risks and Contingencies**

Identifying Test Risks

- The resources required for testing may increase if the testability of the specifications is low.
- The test coverage may be lower than is acceptable if the test case selection strategy is not adequate.
- The test results may not be useful if the correct answers are not clearly specified.

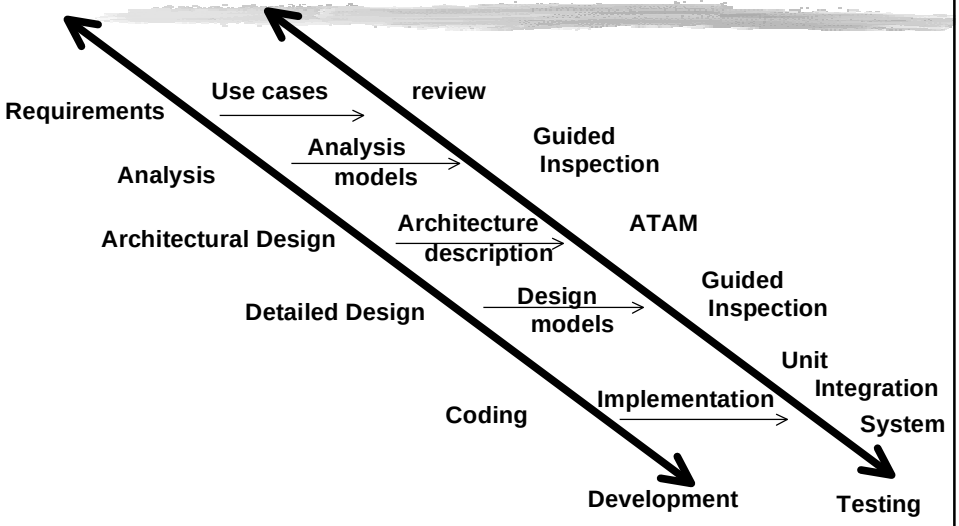
Exercise

- What test activities are done at the product line level in your project?
- What test activities are done at the product level in your project?
- What risks do you see in your testing process?

Software Engineering Activities

- Construct test infrastructure
  - acquire tools that complement developers' tools
  - construct test software
  - mine for test assets
- Execute test plan
  - ATAM
  - Guided Inspection

Process Interactions



Software Engineering Activities

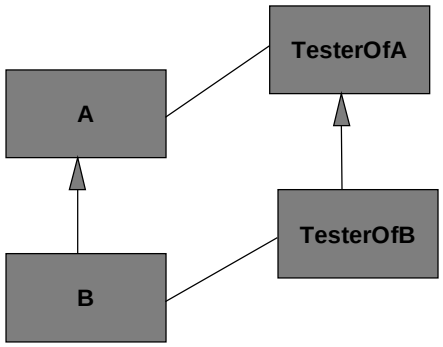
- Encapsulation
  - associate each test asset with a production asset
  - test case is associated with the product it tests
- Information Hiding
  - test interface vs test implementation
- Abstraction
  - follow good design practice
  - test cases are scripts written in a language that supports levels of abstraction

ATAM™

- "Test" the architecture
- Use scenarios from the use cases
- Test for architectural qualities such as
  - extensibility
  - maintainability

(ATAM - Architectural TradeOff Analysis Method is a trademark of the SEI)

"Inheritance" Reuse

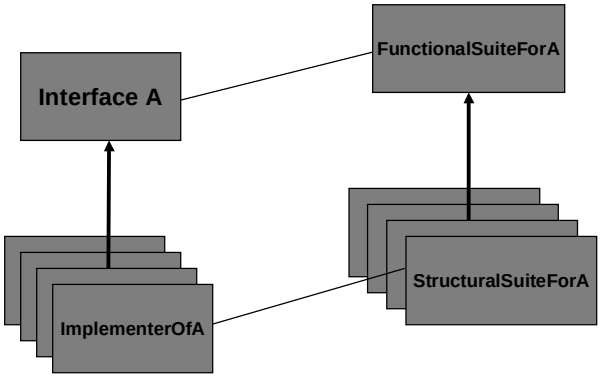


• Test cases created for A are also viable test cases for B

• Test cases for a component are encapsulated in a single component

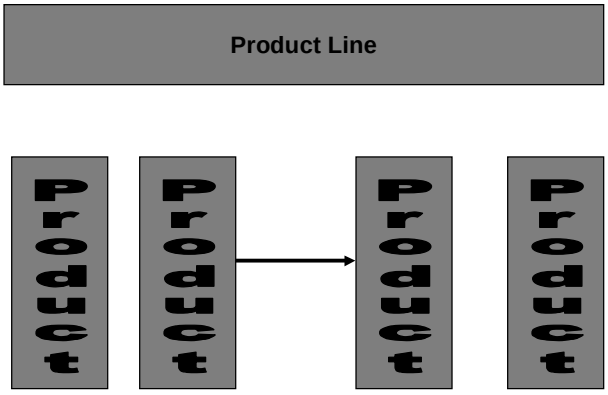
• This inheritance may be:
- **• class inheritance in a programming language**
- **• it may be a specialization relationship in a language without inheritance**
- **• it may be an interface implementation**

Interface Reuse



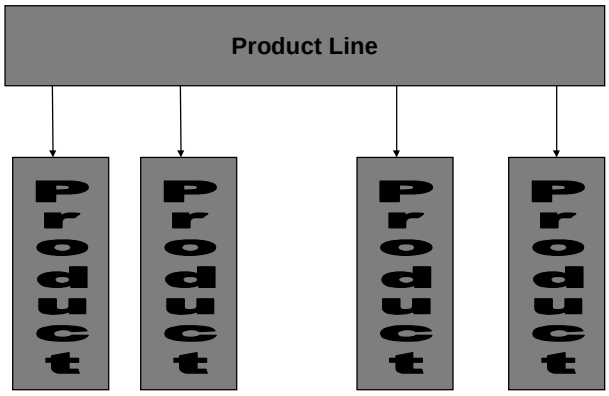
- Functional test cases created for interface A are viable test cases for all implementations of A
- Test cases for a component are encapsulated in a single component
- Structural test cases are defined at the implementation level

Horizontal Test Reuse



- Product assets are reused in other products
- Components may be reused as is from one product to another
- Level of abstraction stays the same
- Test cases will be reused as is

Vertical Test Reuse



- **Product line assets are reused in the individual products**
- **Product line assets include varying levels of abstraction**
- **Abstract assets are specialized for use in products**

Regression Test Strategies

- Regression suite should be a sample of the component, interaction and system test cases.
- Regression testing is triggered by
  - new releases
  - bug fixes
  - transfer across organizational boundaries

Exercise

- List the testing tools currently used by your project.
- How will they contribute to the product line effort?
- How do they need to be modified or improved to better support your product line project?

Product Line Perspective

- Test over complete range of specification since it will be used in many contexts
  - Product line test coverage levels should be higher than at the product level
- Provide test suites for regression testing of assets in a product context
  - Use structures appropriate to the development technology in use

Product Perspective

- What has changed from the product line specification
  - more specific requirements
  - additional requirements
- What has changed from the product line implementation
  - modified components
  - new components

Product Line Test Assets

- Architecture
  - reusable test patterns
- Points of Variability
  - Standard interfaces/standard functional tests
- Regions of Commonality
  - Bundling of test products with development products

Product Line Assets

- Architecture
  - product line - sufficiently general to describe an entire set of products
  - product - sufficiently specific to describe an individual product
- Points of Variability
  - place in the architecture where one product is allowed to be different from other products
- Regions of Commonality
  - place in the architecture where all products must be the same

Architecture

- The product line architecture is verified using a guided inspection process
- The product architectures are also verified using guided inspection sessions
- The test cases for the guided inspection are reused from the product line to product inspection sessions

Points of Variability

- A point of variability is a point in the architecture where products can be different from other products and the product line
- A point of variability is defined by an interface that specifies the services that must be provided by the implementer
- A test suite is defined based on the interface
- Functional tests are passed from product line level to the product level

Regions of Commonality

- A region of commonality is that portion of the architecture that is fixed and remains the same in all products
- These regions are noted only because of their interface with the variable components in the architecture
- The product line implementation of this functionality should be tested in detail once and then only tested for interactions at the product level

Outline

- Overview
- Definitions
- Activities & Assets
- An example company

Example company - ProtocolsRUs

- Builds components that are protocol engines for telecommunication software
- Wants to use a product line approach to produce highly reliable components
- Business plan calls for individual products to take less than 90 days once product line is established

ProtocolsRUs - Test Strategy

- Protocols are constrained by very specific standards and have well-defined states
- The high quality demanded by protocol-based applications requires thorough testing
- Strategies:
  - test from earliest stages of development
  - test sufficiently to achieve 99% reliability

ProtocolsRUs - Test Process & Plan

- Process
  - In addition to the development process-based tests, each of ProtocolsRUs' components must also pass a conformance test
- Plan
  - The plan describes how test assets will be propagated from the product line to the product teams
  - The plan assigns responsibilities to personnel for creation and maintenance

ProtocolsRUs - Test Results & Reports

- Test results and reports are used as the basis for ProtocolsRUs reliability advertising
- Actual reliability measurements are only taken on the final component products
- Each constituent component is also measured as it is developed
- These sub-product measurements are used to estimate the final reliability of the protocol component

Summary Exercise

- Make a list of changes to your testing process that you might recommend.
- Identify whether those changes occur at the product line or product levels.
- What are the trade-offs with each?

Conclusions

- Reuse must occur at all three company levels
- Test plans are templated and "inherited" from the the product line to the individual products
- Product line test cases provide regression test cases for the products
- Test reports provide feedback both to the product team and the product line team

Thanks

- Contact me at:
  - johnmc@korson-mcgregor.com
- Look for us at
  - http://www.korson-mcgregor.com