

Summary of SEI's Architecture Support for Testing initiative

substantial work devoted to this topic in the research community, not much of that research seems to have filtered into communities of practitioners

Categorising Architecture based Testing

Foundations	Concepts, definitions, etc.	
Design	Designing an architecture to make systems more testable	
	Testing whether system will meet its requirements	Testing for conformance of code to architecture
Code present	Testing	Conformance
Code not present	Analysis	N/A

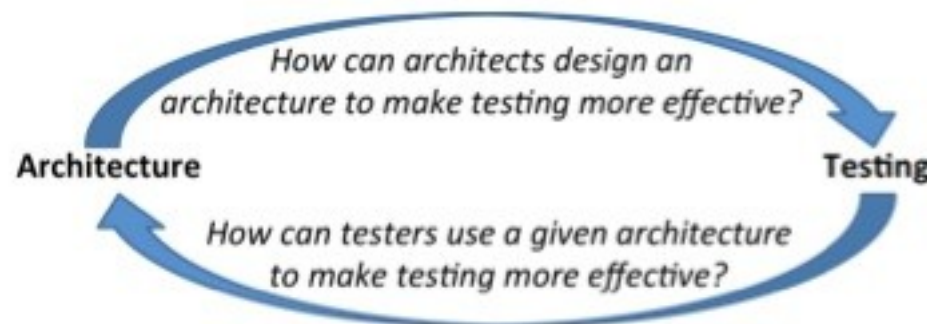
Plan for providing useful techniques for industry

Foundations	awareness of architecture-based testing catalog previous work,
Architecture Analysis	how architecture analysis can lead to improved testing outcomes. <ul style="list-style-type: none">- obviating the need for certain tests,- pointing out the need for other tests,- or reducing the cost of still others.
Testing	workshop for testing practitioners to identify best practices and a set of important model problems in testing.
Conformance	No work currently planned in this area
Design	A catalog of architectural design approaches and anti-approaches that lead to systems that are more testable

Industry Workshop

catalog of architectural design approaches that lead to systems that are more testable.

- patterns, styles, and tactics for testability.
- metrics for object-oriented systems that can help an O-O architect chart a course to a testable system.

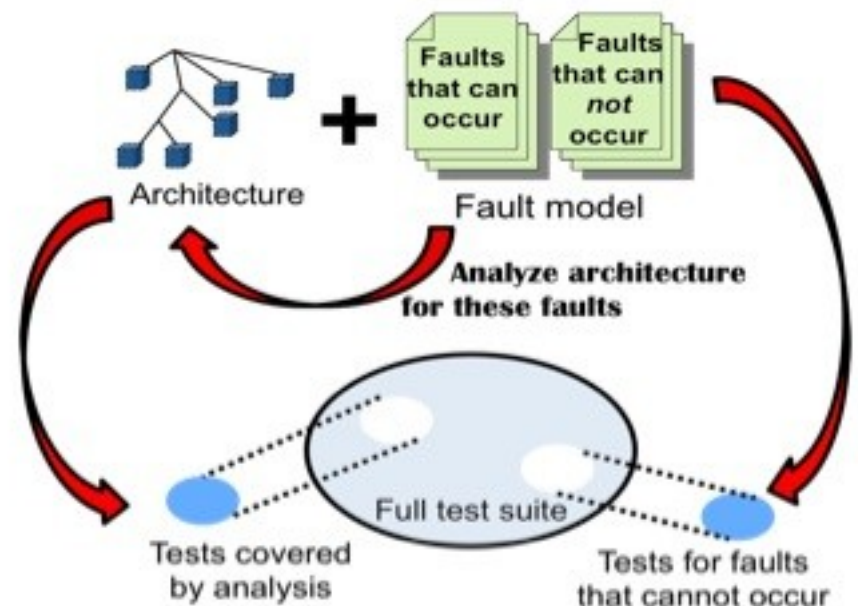


- The catalog will be published soon as an SEI technical note.

Industry Workshop

begun building fault models for architecture design approaches.

- A fault model is a list of faults to which the design approach is both susceptible and immune
- E.g., for pipe-and-filter, SOA based, state-machine



Model problems identified and prioritised

- Architecture, Testing, and Requirements
- Architecture, Testing, and Software Product Lines
- Scope of Architecture and Architecture Support for Testing
- Architecture, Testing, and Integration

Scenario name	REQ1
Source	Tester
Stimulus	A tester chooses a test set to test the system for requirements satisfaction.
Environment	The architecture is complete. System test has not yet begun.
Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of 98% of the requirements. Redundant tests are eliminated.
Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.

Scenario name	INT15
Source	Tester
Stimulus	The tester needs to develop integration tests.
Environment	The architecture of the system is complete and detailed.
Response	Automated tools use the architecture to generate test cases and test subs for the components within the system.
Response measure	The tester asserts that 90% of the generated tests should be run.

Scenario name	REQ2
Source	Tester
Stimulus	A tester chooses a test set.
Environment	The architecture is complete.
Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of the highest-risk areas of the system. Redundant tests are eliminated.
Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.

Scenario name	SPL1
Source	A stakeholder who wants a change to product B.
Stimulus	A change request, which will result in core assets being modified, is initiated for product B, which is a certified product.
Environment	The products in the product line must be certified to a specified standard. Core assets have been used as part of the product implementation. Several products have used these core assets. There is a policy that if any core asset changes, all products that depend on that core asset are rebuilt.
Response	The changes to the core assets are made and the products, A and B, dependent on those assets, are rebuilt using approved tactics. An architecture-based analysis proves that no recertification of A, which only has new implementations of the same core assets, is needed. Evidence accumulated during the changes is used to expedite the recertification of B.
Response measure	The effort to recertify B is significantly less than the effort to certify B originally while maintaining the level of confidence.

Scenario name	INT4
Source	Tester
Stimulus	The tester seeks to uncover performance bottlenecks in the integrated system as quickly as possible.
Environment	The architecture and system integration are complete.
Response	An analysis of the system architecture identifies the components and threads most likely to cause the system to fail in terms of its performance thresholds. The analysis could also be used to predict the components that will cause performance issues when integrated into the system.
Response measure	Performance problems are discovered early and not late in the testing cycle.

Scenario name	INT7
Source	Test designer
Stimulus	The test designer must create a set of tests for system integration.
Environment	The architecture is complete.
Response	The designer uses an automated tool that analyzes the architectural descriptions of the components and generates lists of observable and controllable entities.
Response measure	The tool generates 100% of the observable entities provided by the components within the system.

What are our model problems?

- Can all architecture qualities be captured as BDD tests?
- Can architecture quality BDD tests be included in test automation?
- ...

follow here for more news

- <http://saturnnetwork.wordpress.com/tag/ast-project/>
- WICSA Workshop
 - <http://www.cs.bilkent.edu.tr/ABT-2011/>
- ICSE Workshop
 - <http://ast2011.isti.cnr.it/index.html>
- Active Design Reviews
 - <http://www.sei.cmu.edu/architecture/tools/arid/>

Using a Test-driven mindset to guide the architecture process

**THE PRAGMATIC
ARCHITECT**



Editor: Frank Buschmann
Siemens Corporate Technology,
frank.buschmann@siemens.com

Tests: The Architect's Best Friend

IEEE SW May/June 2011: <http://www.computer.org/portal/web/csdl/doi/10.1109/MS.2011.65>



Whenever I make architecturally significant decisions, I ask myself a simple question:
How can I test my design to ensure that it's sustainable and efficient?

... just taking this viewpoint resulted in architectures that were economical and simpler to understand, realize, and test.

... foremost goal in test-driven design is to establish a strong feedback loop across all phases of software development—to always be in control of the architecture



inception phase:

seeking scenarios that are annotated with concrete quality attributes and relate directly to business-case-relevant features.

scenarios form the very heart of the system and thus provide a harness of “acceptance tests” that guides all my architecture work.

scenarios guide my choices for fundamental architecture styles ... modularization, concepts for key system qualities and ... technologies

scenarios drive an initial architecture review



I pay special attention to designing the interfaces and component interaction.

Design by contract and economic interfaces are a “must.”

I also add special interfaces that provide access to test-relevant component internals.

The use of mock objects and (active design reviews) ADRs to test not-yet-realized interfaces and interactions are, to me, a very powerful form of design simulation.



Equipped with such a safety net, I start the realization of the walking skeleton.

The initial user-level scenarios ensure I'm not losing the big picture; all the other tests guide implementation

When the walking skeleton starts "walking", I start combining the initial scenario tests into a lightweight load and stress smoke test suite. The goal of this suite is to ensure the skeleton has sufficient operational quality to start the next steps in the system's life cycle.



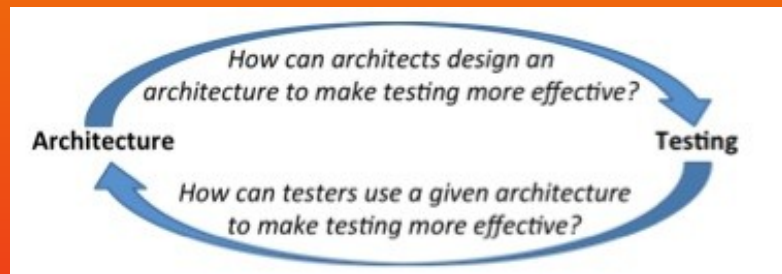
Ideal approach but in reality, I can't realize it to its full extent on some projects, primarily because of organizational and cultural barriers

agile and lean practices can't be dictated.
Patience and endurance are required, along with a step-wise introduction of TDD practices that provide immediate benefit to the project.

It's the architect's responsibility to guide and drive this transition, and most importantly, to practice TDD

Does testing drive your architecture process?

- How do you deal with project management that is not test driven?
- What is the minimum TD-A you can do to provide benefit on an old-fashioned project?
- ...



Summary of SEI's Architecture Support for Testing initiative

- * initiative from SEI
- * SEI is an industrial research org that is funded to improve industry with SW-intensive systems
- * not academic research

<http://www.sei.cmu.edu/>

substantial work devoted to this topic in the research community, not much of that research seems to have filtered into communities of practitioners

@jbaragry

@miles_no

- * lots of academic research but little of it filtering into practice.
- * large potential for improving practice if we can do tech transfer in both directions

Categorising Architecture based Testing

Foundations	Concepts, definitions, etc.	
Design	Designing an architecture to make systems more testable	
	Testing whether system will meet its requirements	Testing for conformance of code to architecture
Code present	Testing	Conformance
Code not present	Analysis	N/A

@jbaragry

@miles_no

Foundations: ..., key concepts, principles for reasoning, theoretical bases, and so forth. It includes what it means for a system to be more testable than another, and how to measure testability.

Architecture Analysis: Architecture-based predictive analysis of systems. This includes static analysis of architectures, model checking of architectural descriptions, simulation of architectures, etc.

Testing: Using architecture to define test assets to test code for functional/quality attribute (QA) correctness. We can use architecture to develop test plans, test paths, test cases for code, or regression tests when architectures or code changes. We can also use architecture to help us define and set up a project's integration and test infrastructure.

Conformance: Using architecture to define tests for code compliance/conformance to architecture.

Design: Designing an architecture to make systems more testable. This work includes finding and cataloging patterns and tactics for testability, designing testing as a service, and capturing anti-patterns: what not to do in the architecture if you'd like a testable system.

Plan for providing useful techniques for industry

Foundations	awareness of architecture-based testing catalog previous work,
Architecture Analysis	how architecture analysis can lead to improved testing outcomes. - obviating the need for certain tests, - pointing out the need for other tests, - or reducing the cost of still others.
Testing	workshop for testing practitioners to identify best practices and a set of important model problems in testing.
Conformance	No work currently planned in this area
Design	A catalog of architectural design approaches and anti-approaches that lead to systems that are more testable

@jbaragry

@miles_no

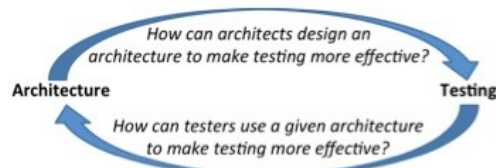
How is this project focussing its efforts

<http://saturnnetwork.wordpress.com/2010/11/23/the-sei-archi>

Industry Workshop

catalog of architectural design approaches that lead to systems that are more testable.

- patterns, styles, and tactics for testability.
- metrics for object-oriented systems that can help an O-O architect chart a course to a testable system.



- The catalog will be published soon as an SEI technical note.

@jbaragry

@miles_no

Industry workshops and research workshops happening.

For industry:

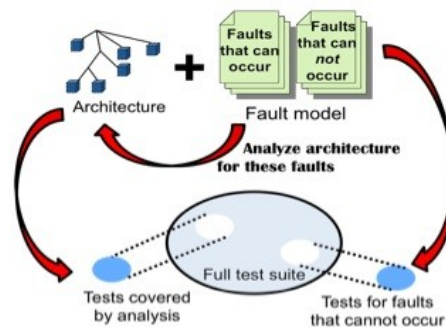
Want:

- * What are archs doing to help testers do their jobs
- * What model problems if they were solved, would make a substantial improvement in how architecture and testing can be done.

Industry Workshop

begun building fault models for architecture design approaches.

- A fault model is a list of faults to which the design approach is both susceptible and immune
- E.g., for pipe-and-filter, SOA based, state-machine



@jbaragry

@miles_no

E.g., SOA fault model

A service violates its contract.

It accepts/delivers values outside the bounds defined for the service.

It fails to meet performance expectations.

A component containing the service fails, as observed by termination of a process.

A node containing the component fails as observed by inability to reach the node.

FSM immune:

if your system doesn't shortcut that logic anywhere else, then you can be assured that system-wide control logic errors can occur only in the state machine.

Testers can concentrate more on faults that can occur and less on faults that cannot. In extreme cases, analysis that shows with 100% certainty that a particular fault cannot occur might take the place of corresponding tests

<http://saturnnetwork.wordpress.com/2011/02/22/architecture-support-for-testing-an-update/>

Model problems identified and prioritised

- Architecture, Testing, and Requirements
- Architecture, Testing, and Software Product Lines
- Scope of Architecture and Architecture Support for Testing
- Architecture, Testing, and Integration

@jbaragry

@miles_no

The output of that workshop was a set of 30 model problems
if solved, would result in a significant decrease in project resources devoted to testing and/or a significant increase in system quality given an expenditure level

Grouped into 4 main areas.

Then voted on by workshop participants and open vote after publishing on their blog.

Fed to the researcher workshop to get researchers to focus on things that industry really needs.

Scenario name	REQ1
Source	Tester
Stimulus	A tester chooses a test set to test the system for requirements satisfaction.
Environment	The architecture is complete. System test has not yet begun.
Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of 98% of the requirements. Redundant tests are eliminated.
Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.

@jbaragry

@miles_no

Most popular

Can we use the architecture to tell us that (out of all of the huge number of tests possible)

if we run a small number of tests, we will have covered 98% of the requirements?

Scenario name	INT15
Source	Tester
Stimulus	The tester needs to develop integration tests.
Environment	The architecture of the system is complete and detailed.
Response	Automated tools use the architecture to generate test cases and test subs for the components within the system.
Response measure	The tester asserts that 90% of the generated tests should be run.

@jbaragry

@miles_no

Next most popular

How can architecture be used to drive integration testing? Generation of test stubs from high-level descriptions (Common Information Model CIM)

Generation of test cases from more detailed descriptions (Multispeak)

Keyword-based test automation—use of domain-specific language. Can we use the domain-specific language in the architecture models so that we can test sequences of the keywords... basis for auto-generation of tests

Scenario name	REQ2
Source	Tester
Stimulus	A tester chooses a test set.
Environment	The architecture is complete.
Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of the highest-risk areas of the system. Redundant tests are eliminated.
Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.

@jbaragry

@miles_no

4-way tie for 3rd

Can I use architecture to use tests to cover high-risk or high-probability-of-error areas in the system?

Design for testability... of the architecture to meet those requirements (encapsulation, isolation...)

Predictive analysis... to analyze for satisfaction of those reqs.

Generation of test artifacts... to test for those requirements

Conformance... is necessary for the other three to work.

Scenario name	SPL1
Source	A stakeholder who wants a change to product B.
Stimulus	A change request, which will result in core assets being modified, is initiated for product B, which is a certified product.
Environment	The products in the product line must be certified to a specified standard. Core assets have been used as part of the product implementation. Several products have used these core assets. There is a policy that if any core asset changes, all products that depend on that core asset are rebuilt.
Response	The changes to the core assets are made and the products, A and B, dependent on those assets, are rebuilt using approved tactics. An architecture-based analysis proves that no recertification of A, which only has new implementations of the same core assets, is needed. Evidence accumulated during the changes is used to expedite the recertification of B.
Response measure	The effort to recertify B is significantly less than the effort to certify B originally while maintaining the level of confidence.

@jbaragry @miles_no

Testing Certified Product Line Assets : Can AST help to reduce certification costs for products by identifying and managing certifications levels of core assets?

Change core or variant assets for next Product B without affecting certification of Product A

Cost is an issue

Risk to product delivery

Takes too much time to perform testing for delivery

What does it take to certify a parent product having capability and then apply certification to other products?

Product A has completed the certification and a problem is found in a separate product. How can we prevent certification process if Product A needs to pick up change?

Does early or late binding help in reducing certification costs?

Scenario name	INT4
Source	Tester
Stimulus	The tester seeks to uncover performance bottlenecks in the integrated system as quickly as possible.
Environment	The architecture and system integration are complete.
Response	An analysis of the system architecture identifies the components and threads most likely to cause the system to fail in terms of its performance thresholds. The analysis could also be used to predict the components that will cause performance issues when integrated into the system.
Response measure	Performance problems are discovered early and not late in the testing cycle.

@jbaragry

@miles_no

Can the architecture be used to predict and define the bottlenecks within the system and then be used as a basis for integration testing?

Utilization and resource availability can be bottleneck predictors.

Capturing sufficient performance information to determine high transaction components

Predicting integration risk

Scenario name	INT7
Source	Test designer
Stimulus	The test designer must create a set of tests for system integration.
Environment	The architecture is complete.
Response	The designer uses an automated tool that analyzes the architectural descriptions of the components and generates lists of observable and controllable entities.
Response measure	The tool generates 100% of the observable entities provided by the components within the system.

@jbaragry

@miles_no

How can a module definition describe what can be observed?

Not really sure what this means

What are our model problems?

- Can all architecture qualities be captured as BDD tests?
- Can architecture quality BDD tests be included in test automation?
- ...

@jbaragry

@miles_no

Can we brainstorm on this?

What sort of problems if solved would help you?

Open Space?

follow here for more news

- <http://saturnnetwork.wordpress.com/tag/ast-project/>
- WICSA Workshop
 - <http://www.cs.bilkent.edu.tr/ABT-2011/>
- ICSE Workshop
 - <http://ast2011.isti.cnr.it/index.html>
- Active Design Reviews
 - <http://www.sei.cmu.edu/architecture/tools/arid/>

Using a Test-driven mindset to guide the architecture process

**THE PRAGMATIC
ARCHITECT**



Editor: Frank Buschmann
Siemens Corporate Technology,
frank.buschmann@siemens.com

Tests: The Architect's Best Friend

IEEE SW May/June 2011: <http://www.computer.org/portal/web/csdl/doi/10.1109/MS.2011.65>

@jbaragry

@miles_no

MAY/JUNE 2011 | IEEE SOFTWARE



Whenever I make architecturally significant decisions, I ask myself a simple question:
How can I test my design to ensure that it's sustainable and efficient?

... just taking this viewpoint resulted in architectures that were economical and simpler to understand, realize, and test.

... foremost goal in test-driven design is to establish a strong feedback loop across all phases of software development—to always be in control of the architecture

Start with the test-driven mindset to get a better result

Following the process creates a constant feedback loop that allows adaptation underway



inception phase:

seeking scenarios that are annotated with concrete quality attributes and relate directly to business-case-relevant features.

scenarios form the very heart of the system and thus provide a harness of “acceptance tests” that guides all my architecture work.

scenarios guide my choices for fundamental architecture styles ... modularization, concepts for key system qualities and ... technologies

scenarios drive an initial architecture review

Start with identifying scenarios

Scenario are related to quality attributes and business relevant features

Acceptance test qualities driven the initial “Rough upfront Design”

This can then feed into the first architecture review where neutral experts determine how the arch can support those scenarios.

Refactoring ensues



I pay special attention to designing the interfaces and component interaction.

Design by contract and economic interfaces are a "must."

I also add special interfaces that provide access to test-relevant component internals.

The use of mock objects and (active design reviews) ADRs to test not-yet-realized interfaces and interactions are, to me, a very powerful form of design simulation.

Active design reviews are an effective technique for ensuring quality, detailed designs in software [Parnas 85].

The method relies on actively engaging the reviewers by assigning them review tasks that are carefully structured to avoid asking yes/no questions.

Instead of:

Is the performance of each service adequately specified?

Use:

For each service, write down its maximum execution time and list the shared resources that it may consume.

THE PRAGMATIC ARCHITECT



Editor: Frank Buschmann
Siemens Corporate Technology,
frank.buschmann@siemens.com

Equipped with such a safety net, I start the realization of the walking skeleton.

The initial user-level scenarios ensure I'm not losing the big picture; all the other tests guide implementation

When the walking skeleton starts "walking", I start combining the initial scenario tests into a lightweight load and stress smoke test suite. The goal of this suite is to ensure the skeleton has sufficient operational quality to start the next steps in the system's life cycle.

@jbaragry

@miles_no

For walking skel ... I use additional test tools where possible: profilers to check behavior, code quality management tools to support idiomatic programming and avoid architecture drift, pair programming to guide coding, and more architecture reviews by experts

I create additional tests to guide any needed adjustment

Next phase: whatever they are: construction, integration, handover to integration and system test, bug fixing, deployment, maintenance, evolution



Ideal approach but in reality, I can't realize it to its full extent on some projects, primarily because of organizational and cultural barriers

agile and lean practices can't be dictated. Patience and endurance are required, along with a step-wise introduction of TDD practices that provide immediate benefit to the project.

It's the architect's responsibility to guide and drive this transition, and most importantly, to practice TDD

Does testing drive your architecture process?

- How do you deal with project management that is not test driven?
- What is the minimum TD-A you can do to provide benefit on an old-fashioned project?
- ...

@jbaragry

@miles_no

Can we brainstorm on this?

What is the minimum TDArchitecture we can do in situations where the customer isn't test driven?