The Role of Software Architecture

Rich Reitman

at Cornell University

February 14, 2008

Adobe

The Role of Software Architecture - Agenda

- What is software architecture?
- How do we capture software architecture?
- What do software architects do?

My Context

- Commercial software companies
  - Rational Software: 15 → 4000
  - IBM: 300,000
  - Adobe Systems: 6000
- Multiple roles (primarily technical or technical management)
  - Lead Developer, Product Architect
  - Software engineering manager
  - Chief software architect
  - General Manager, CTO
  - Mentor to software architects

What is Software Architecture?

From IEEE 1471-2000:

Software architecture is the fundamental organization of a system

- embodied in its components,
- their relationships to each other and the environment,
- and the principles governing its design and evolution

What is Software Architecture?

From Unified Process (Jacobson, Booch, Rumbaugh):

Software architecture encompasses the set of significant decisions about the organization of a software system

- Selection of the structural elements and their interfaces
- Behavior as specified in collaborations among those elements
- Composition of these structural and behavioral elements into larger subsystems
- Architectural style that guides this organization

What is Software Architecture?

- All software systems have an architecture
- Even if:
  - It isn't written down
  - No one understands it
  - There are no architects
- How are we going to control and evolve the architecture?
- How do we capture software architecture?

Capturing Software Architecture

- Problem
- Solution
- Architectural Quality

- Benefits
  - Communication tool
  - Identify and Address system level risks

Capturing Software Architecture - Problem

- Identify architecturally significant requirements
  - Actors
  - Use cases
  - Non-functional
  - Constraints

- Use cases vs. User interfaces vs. User Experience

Example - “Peripheral Actors” key to adoption

- Problem: provide a way of controlling the sharing of PDF files
  - Includes posting on the web, sending email, etc.
- Solution: provide server
  - Supports both simple and complex policies
  - PDF Reader operation confirms with server
- How is the system managed? (backup/recovery, load balancing, …)
  - Early on: different (better?) methods
  - Now: can use standard methods from major vendors (DB, J2EE app server, …)
- Observation: be sure that all actors are considered
  - The (peripheral) maintainer actor was key to adoption
  - Cost of operation vs. benefit of capability

Capturing Software Architecture - Solution

- Design a solution - 4 +1 views
  - Use Case - provides behavior
  - Logical - realization of functional requirements
  - Implementation
  - Deployment
  - Process
  - [Data - additional view used at Adobe]
- Focus on architecturally significant aspects
- CMU/SEI - many views; pick the ones you need

Example - Plug & Play for Military Ships

- CelsiusTech
  - Building the same sorts of ships repeatedly
  - Define a software architecture based on abstraction of ship systems
  - New hardware physically plugs in: radar, weapons, etc.
  - Corresponding software plugs into the software architecture (object-oriented)
- Business-Driven Decision
  - Reduced cost & risk
  - 65% reuse between Danish and Swedish ships
- CelsiusTech business turn around

Capturing Software Architecture - Architectural Quality

- System Characteristics - realize non-functional requirements
  - Reliability, Security, Availability, …
  - Performance, Scalability, …
  - Testability, Maintainability, Extensibility, …
  - Usability, Localization, …
  - …

Example - Incremental Compilation

- Problem: changing software interfaces causes massive recompilation
- Solution: extend syntax-directed paradigm to limit recompilation
  - Determine impact at a granularity much finer than file
  - Apply technique recursively to determine all places of possible impact
  - Treat affected areas as non-terminal nodes associated with text needed compilation
- Quality characteristics
  - Testability - full compilation and incremental compilation should yield same results
    - Randomly generate sequence of changes & confirm
    - Output seed along with failure to enable reproduction
  - Usability
    - 1st release: syntax tree model explicit to user - best precision
    - 2nd release: underlying model invisible - everyone uses it!

What do software architects do?

- Define the architecture
- Maintain the architectural integrity of the system
- Assess technical risks & find risk mitigation strategies
- Propose order and content of development iterations
- Consult on design, implementation, integration, test
- Participate in determining future system directions

- Time allocation rule of thumb
  - 50% architecting: designing, prototyping, documenting
  - 25% getting input: users, requirements, other architectures & technologies
  - 25% providing info: communicating the architecture, assisting

Based on "What do software architects do?", by Philippe Kruchten

Software Architecture and Agile Development Methods

- Agile Methods advocate avoiding “Big Design Up Front”
  - Get close to customers with working code and iterate
  - Enabled by: test-driven development, continuous integration & refactoring
- Agile Architectural Focus (architect =? coach)
  - Where is there risk? - Prototype and focused reviews
  - What are the recurring patterns and paradigms? - Refactor
  - What are the system-level characteristics? - Design tests
  - Enable the team - communicate
  - Less is more - Document only what isn’t captured easily in code
- Architecture and Agility are complementary

Observation: Architecture is a Social Process

Technology does not exist in a vacuum

- It's built by people.
- It's used by people.
- It's supported by people

The goal of good architecture is social understanding

- The structure is well understood by the engineering community
- The capabilities are well understood by the product management community
- The usage is well understood by the user community

Corollary: Architects must know their community

Social processes must adapt to their scale

- For small projects, scrum can be ideal
- For large projects, you need a clearly articulated process and good infrastructure
- Architects must adapt to and help shape the community practice

Social processes are only effective if they are voluntary

- Everyone must agree on the goals, and follow the rules
- Everyone must trust that all are operating with trust and integrity
- Architects can only lead if they are trusted to listen and learn

Questions

And

Discussion

Adobe

Revolutionizing
how the world engages
with ideas and information

Adobe