SWEN30006
Software Modelling and Design

ARCHITECTURAL ANALYSIS

Larman Chapter 33

Error, no keyboard - press F1 to continue.

—early PC BIOS message

Objectives

On completion of this topic you should be able to:

- Identify architecturally significant requirements
- Create architectural factor tables to capture factors that influence architectural decisions
- Create technical memos that record architectural decisions

Architectural Analysis

- concerned with identifying and resolving system's non-functional requirements in the context of its functional requirements.
- includes identifying and analysing:
  - architecturally significant requirements
  - variation points
  - probable evolution points
- reduces risk of missing a critical factor in system design, and helps focus effort

Arch. Significant Functional Req'nts: Examples

Function	Description
Auditing	Provide audit trails of system execution.
Licensing	Provide services for tracking, acquiring, installing, and monitoring license usage.
Localization	Provide facilities for supporting multiple human languages.
Mail	Provide services that allow applications to send and receive mail.
Online help	Provide online help capability.
Printing	Provide facilities for printing.
Reporting	Provide reporting facilities.
Security	Provide services to protect access to certain resources or information.
System management	Provide services that facilitate management of applications in a distributed environment.
Workflow	Provide support for moving documents and other work items, including review and approval cycles.

Capturing Architectural Requirements

Arch. Significant Non-Functional Requirements

- *Usability*
  - e.g. aesthetics and consistency in the UI.
- *Reliability*
  - e.g. availability (the amount of system "up time"), accuracy of system calculations, and the system's ability to recover from failure.
- *Performance*
  - e.g. throughput, response time, recovery time, start-up time, and shutdown time.
- *Supportability*
  - e.g. testability, adaptability, maintainability, compatibility, configurability, installability, scalability, and localizability.

Capturing Architectural Requirements

Identification &Resolution: Examples

1. effect of reliability/fault-tolerance req'nts on design?
   - remote services failover to local?
2. effect of re-use components licencing costs on profit?
   - excellent DB server for 2% of sales?
3. effect of adaptability/configurability req'nts on design?
   - business rule variations: user customise, or charge them
4. effect of brand name and branding on architecture?
   - protected variation: name, logos, icons, …

Common Steps in Architectural Analysis

Steps (occurs in early elaboration):

1. Identify/analyse *architectural factors:* requirements with impact on the architecture (esp. non-functional)
   - overlaps with requirements analysis
   - some identified/recorded during inception, now investigated in more detail
2. For the architectural factors, analyse alternatives and create solutions: *architectural decisions*
   - *e.g. remove requirement; custom solution; stop project; hire expert*

Priorities

1. Inflexible constraints
   - e.g. safety; security, legal compliance
2. Business goals
   - e.g. demo for clients: tradeshow in 18 months
   - e.g. competitor-driven window-of-opportunity
3. Other goals
   - requirements all relate back to higher-level goals, and eventually to business goals
   - e.g. extendible → new release every 6 months

Architectural Factor Table

1. Factor
2. Measures and quality scenarios
3. Variability (current flexibility and future evolution)
4. Impact of factor (and its variability) on stakeholders, architecture and other factors
5. Priority for success
6. Difficulty or risk

1. Recovery from remote service failure

2. When remote service fails, re-establish connectivity with it within 1 min. of its detected re-availability, under normal store load in a production environment.
3. *current flexibility:* SME—local client-side simplified services acceptable (& desirable) until reconnect is possible.

   evolution: within 2 years, some retailers may be willing to pay for full local replication of remote services (such as the tax calculator). Probability? High.
4. High impact on large-scale design. Retailers really dislike it when remote services fail, as it prevents them from using a POS to make sales.
5. Priority: H
6. Difficulty or Risk: M

Technical Memo

1. Issue
2. Solution Summary
3. Factors
4. Solution
5. Motivation
6. Unresolved Issues
7. Alternatives Considered

1. Reliability-Remote Service Failure Recovery

2. Location transparency using service lookup, failover from remote to local, partial local replication
3. a. Robust recovery from remote service failure

   b. Robust recovery from remote product DB failure

   e.g. (a) tax calculator (b) prices and descriptions
4. Achieve *protected variation* w.r.t. location of services using an *adaptor …*
5. Retailers don't want to stop making sales! …
6. None 7. "gold level" credit authorisation too costly

Separation of Concerns

Applies at architectural level as well as small scale.

Cross-cutting concerns:

- those with a wide application or influence across the system, e.g. data persistence, security

Example techniques:

1. Modularize into separate component/subsystem
2. Use decorators: container that wraps inner object
3. Use post-compilers or aspect-oriented technology

Summary of Themes in Arch. Analysis

1. Concerns are especially related to non-func. requirements with awareness of business context, but addressing func. requirements and their variability.
2. Concerns involve system-level, large-scale, broad problems, with resolution involving large-scale or fundamental design decisions.
3. Must address interdependencies and trade-offs (e.g. security/performance, or anything/cost)
4. Involves the generation/evaluation of alternatives.