Metrics for Sustainable Software Architectures
An Industry Perspective
Agenda

- Introduction
- Goal-Question-Metrics method
- GQM methodology to identify metrics for sustainable software architectures
- Measurement process
- Conclusions
**Introduction**

- Industry practitioners grapple with poor and degraded architectures that slow down the progress of software evolution and reduce profits.
- Practitioners require architecture metrics that support sustainable software architectures.
- What are the best metrics for tracking and enhancing architectures?
  - It depends. Metrics evolve specific to the industry and application domain.
  - Domain concepts like:
    - The business principles for the organization
    - The quality attributes related to these business principles
- Can help refine the solution space.
Principles

- Are there principles that can be used to select between alternative metrics?
- For example:
  - Metrics should be as simple and transparent as possible,
  - Metrics should be tied to business value
  - Etc.
Principles Discussion

- What principles do you consider important for architecture sustainability?
Quality Attributes

- Prioritized quality attribute scenarios and their response measures related to principles are a good place to start in our search for metrics.

- Quality Attributes such as:
  - maintainability
  - extensibility
  - reliability
  - modularity

- Each of these areas relates to principles that provide value to the product and the organization
  - We focus on the business value principle in this walkthrough
Business value for sustainable architecture

Quality Attributes

- Extensibility
- Maintainability
- Reliability
- Modularity

Business Value

Cost ($) of developing a new feature
- Development cost
- Integration cost (modifying existing code)
- Testing cost

Cost ($) of fixing bugs
- Per component development cost
- Testing and regression testing cost
Goal-Questions-Metrics method

- GQM is a framework for developing a metrics program
- GQM steps include:
  - Generate a set of organizational goals
  - What do you want to improve?
  - Derive a set of questions relating to the goals
  - Answers provide visibility into meeting the goals
  - Develop a set of metrics needed to answer the questions
GQM Definitions

- Define major goals behind evaluating the sustainability of a software architecture using quality attributes

- Questions derived from goals that must be answered to determine if the goals are achieved

- Measurements that provide the most appropriate information for answering the identified questions
Sustainability of the architecture of a software system

- Identify simple ways to measure the sustainability of a software architecture that considers metrics that measure maintainability, extensibility, reliability, and modularity
- Metrics should be connected to economic benefits
- Metrics should be as simple as possible
- Collection of metrics should be as transparent as possible
Maintainability

- “The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment” (IEEE)

- “Maintainability is the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changing environment” (Aldekoa, et al.)

- Improving **software maintainability** thorough architecture design involves guiding the dependencies of architecture components to their most optimal configuration.
Maintainability

GQM

Goal

Reduce the effort to incorporate new features in the software

Questions

How much effort does it take to develop a new feature?

How much effort does it take to integrate a new feature?

Metrics

Person-hours effort needed to develop and unit-test a new feature

Person-hours effort needed to integrate and test a new feature

Business Value

Cost ($) of developing a new feature

Cost ($) of integrating a new feature

How much time does it take to fix defects?

Person-hours effort needed to fix defects

Person-hours effort needed to test bug fixes

Cost ($) of fixing bugs

Cost ($) of testing bug fixes and doing regression tests
Extensibility

- Software extensibility is the ability that a software system has to be extended and the level of effort required to implement the extension.
- Extensions are additions of new functionality, or modifications to existing functionality.
- The main issue is to provide enhancements to the software while minimizing impact to existing system functions.
- From the software architecture perspective extensibility means the system is designed to include “hooks” and mechanisms for expanding and enhancing the system without having to make major changes to the system architecture.
- Extensibility can also mean that the behavior of the software system can be modifiable at run time, without having to recompile or change the original source code.
- Not easy to find meaningful way to measure extensibility.
Extensibility

GQM

Goal
Reduce the effort to develop totally new functionality
Reduce the effort to make modifications to existing functionality

Questions
How much effort does it take to develop fundamentally new features?
How much time does it take to make modifications to existing functionality?

Metrics
Person-hours effort needed to develop and unit-test a totally new feature
Person-hours effort needed to integrate and test a totally new feature
Person-hours effort needed to develop and unit-test modifications
Person-hours effort needed to integrate and test modifications

Business Value
Cost ($) of developing a new feature
Cost ($) of integrating a new feature
Cost ($) of coding modifications to existing functionality
Cost ($) of integrating and testing modifications
The reliability of a system is a measure of its ability to provide a failure-free operation.

For many practical situations, reliability of a system is represented as the failure rate.

We wish to apply reliability to software architecture by assessing the reliability on a component basis when possible and determining the impact of modularity on reliability.
Reliability
GQM

Goal

Do the product and product components meet reliability targets and expectations?

Questions

Number of software attributable failures reported per installation during the first system year of customer use

Number of reported failures attributed to each software component during the first system year of customer use

Metrics

Cost ($) associated fixing functional defects found by customers

Maintenance Cost ($) per architecture component

Business Value

Improve customer Satisfaction on product usage

© ABB
Month DD, YYYY | Slide 16
Modularity

- Modularity refers to the degree that a system has to allow its components to be separated or recombined.

- Software modularity refers to the logical partitioning of components or modules that allows complex software to be manageable for the purpose of development, implementation and future maintenance.

- The logic of partitioning of modules may be based on related functions, implementation considerations, data needs, etc.
Modularity

GQM

Reduce the extent to which a change in one component impacts other components of the software

Goal

Questions

Metrics

Business Value

How much does a change to a component impact other components?

Degree of impact – number of components requiring change or regression test with change to a subject component.

Cost ($) of fixing defects in each component

How many defects are injected to components when a particular component is modified?

Number of defects injected to other components when modifying a particular component

Cost ($) of fixing defects in other components

How much time does it take to fix defects when modifying a component?

Person-hours effort needed to fix defects in other components

Cost ($) of fixing defects in other components

Questions

Metrics

Business Value

© ABB
Month DD, YYYY | Slide 18
Measurement process

- Define Measurement Goals/Objectives
- Specify questions
- Define metrics
- Specify data collection mechanisms
- Identify data storage procedures
- Specify data analysis procedures
- Provide measurement results
Conclusions

- Degraded architectures slow down the progress of software evolution and reduce profits
- Industry requires architecture metrics that support sustainable software architectures
- Proposed quality attributes that can serve as a basis for sustainable architectures, include:
  - Maintainability
  - Extensibility
  - Reliability
  - Modularity
- GQM is a methodology that can be utilized to derive metrics for software architecture sustainability
- Metrics should be simple and easy to collect
- Metrics should be linked to economic benefits