CPSC 875

John D. McGregor

C 23 – Wrap-up

Role/Process/Workproduct

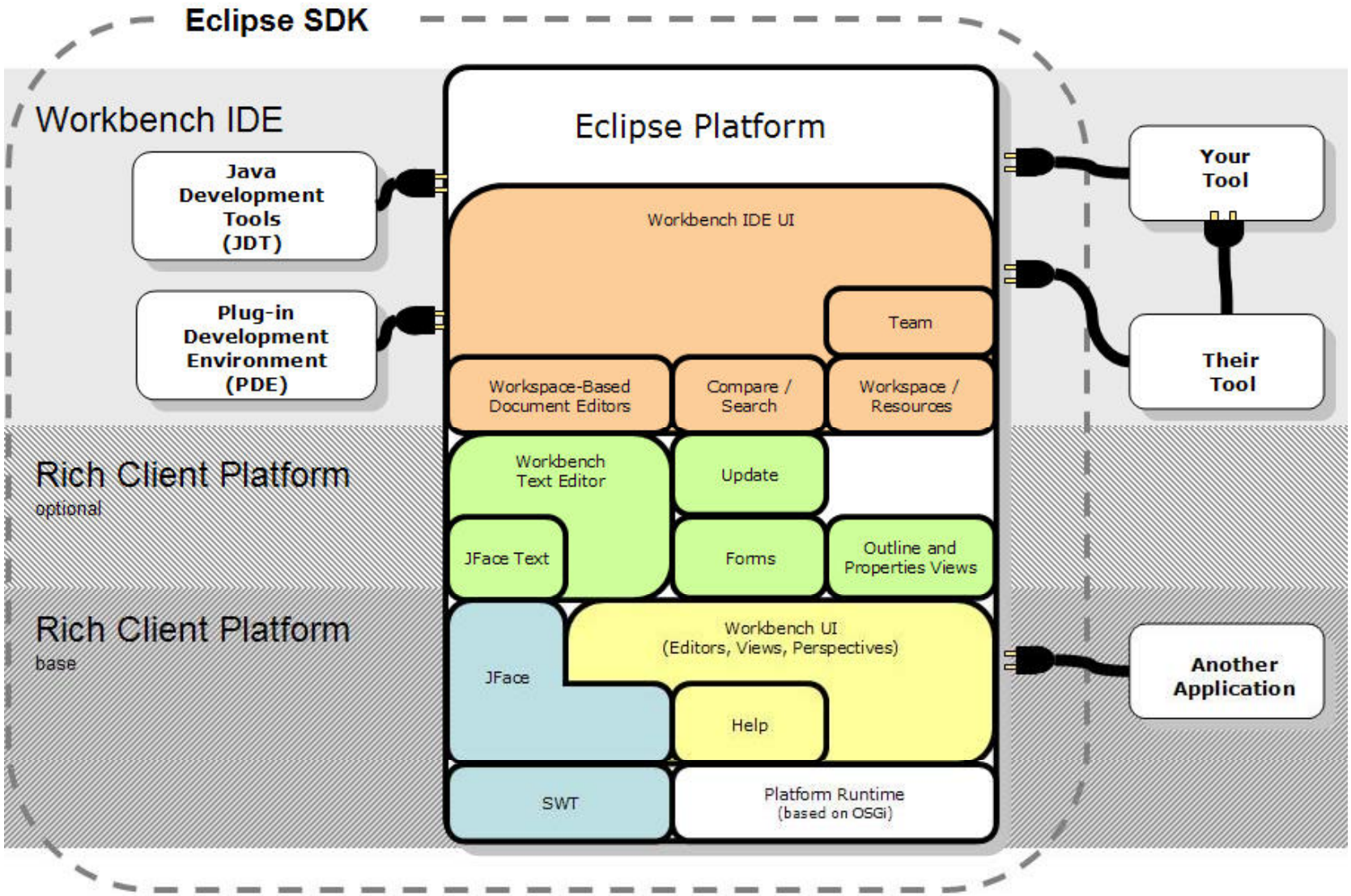
- The architect
  - What is their place in the organization
  - How do they earn their keep?
- The architecture process
  - What are the different processes?
  - What is the value of each?
- The architecture
  - What is its purpose?
  - How is it used?

Definition

- The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

Software Architecture in Practice (2nd edition), Bass, Clements, Kazman; Addison-Wesley 2003

Eclipse architecture cartoon



Pieces

- Modules, subsystems, …
- These are pieces of a system and entities with which the architect works.
- Determining what a particular module should do is the job of the architect
- How a particular module does its assigned job is a detailed design issue not an architecture issue
- Architectural issues cross module boundaries

Baldwin's Operators

- Splitting
- Substituting
- Augmenting
- Excluding
- Inverting
- Porting

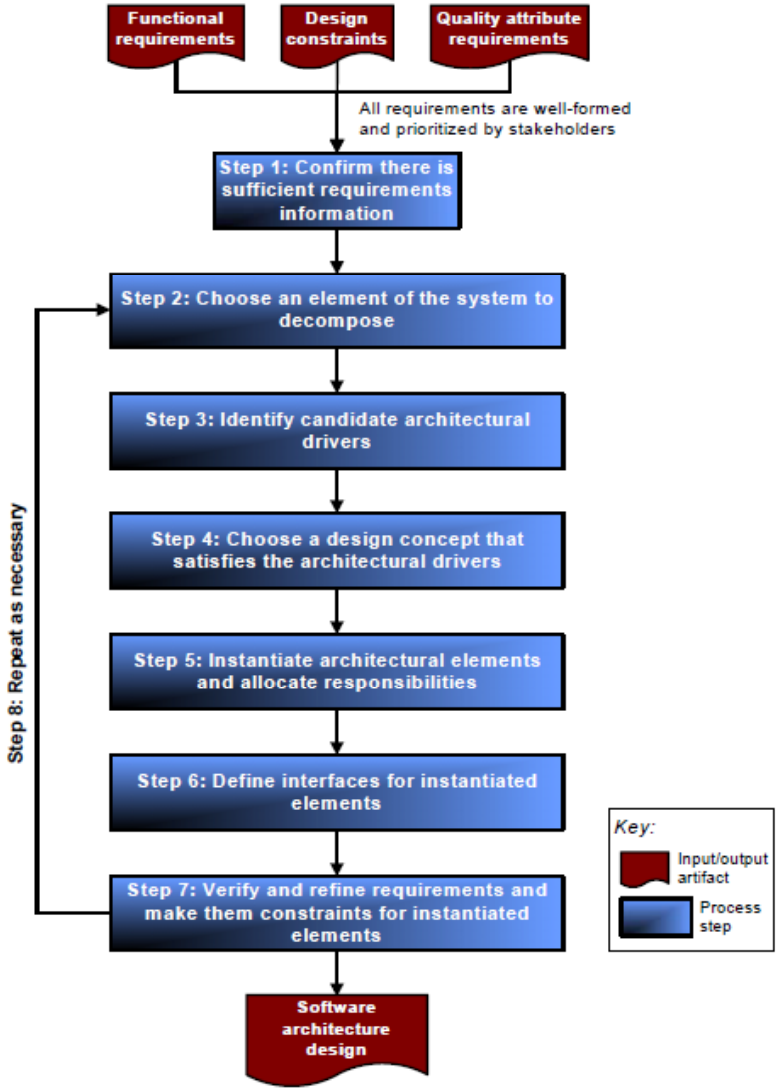
Orchestration/choreography

- The architect creates pieces for certain reasons
- And connects certain pieces to achieve specific objectives.
- The architect orchestrates the interactions of the pieces of the system but leaves the details to the engineers.

Functional/non-functional

- If all we cared about were functional requirements then no structure would be needed. The box below would be the product and the architecture.

Steps


Functional requirements
Design constraints
Quality attribute requirements
All requirements are well-formed and prioritized by stakeholders
Step 1: Confirm there is sufficient requirements information
Step 2: Choose an element of the system to decompose
Step 3: Identify candidate architectural drivers
Step 4: Choose a design concept that satisfies the architectural drivers
Step 5: Instantiate architectural elements and allocate responsibilities
Step 6: Define interfaces for instantiated elements
Step 7: Verify and refine requirements and make them constraints for instantiated elements
Step 8: Repeat as necessary
Software architecture design
Key:
Input/output artifact
Process step

Who determines priorities

- Business goals – set by a dictator or by a consensus building process set a high-level direction
- Stakeholders
  - Users
  - Customers
  - Suppliers
  - Developers
  - Testers
  - Etc.

Quality attributes

- IEEE Std. 1061 subfactors:

Efficiency
- **Time economy**
- **Resource economy**

Functionality
- **Completeness**
- **Correctness**
- **Security**
- **Compatibility**
- **Interoperability**

Maintainability
- **Correctability**
- **Expandability**
- **Testability**

Portability
- **Hardware independence**
- **Software independence**
- **Installability**
- **Reusability**

Reliability
- **Non-deficiency**
- **Error tolerance**
- **Availability**

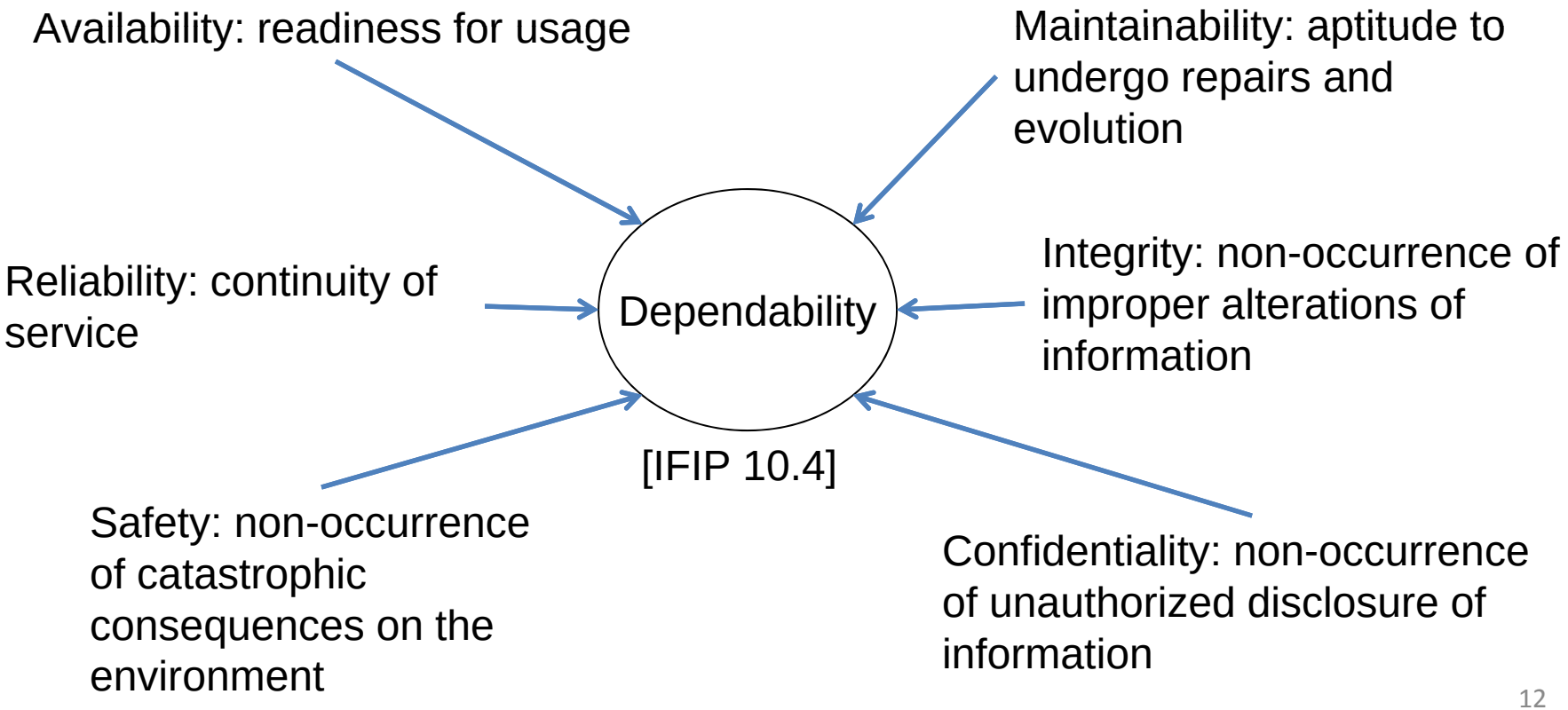
Usability
- **Understandability**
- **Ease of learning**
- **Operability**
- **Comunicativeness**

http://en.wikipedia.org/wiki/ISO/IEC_9126

Qualitative/Quantitative

Definition of Dependability

Dependability is the ability of a system to deliver service that can justifiably be trusted [63]



Styles and patterns

- An architecture style and a pattern are very similar
- A pattern may have more information, particularly more information about trade-offs among attributes.

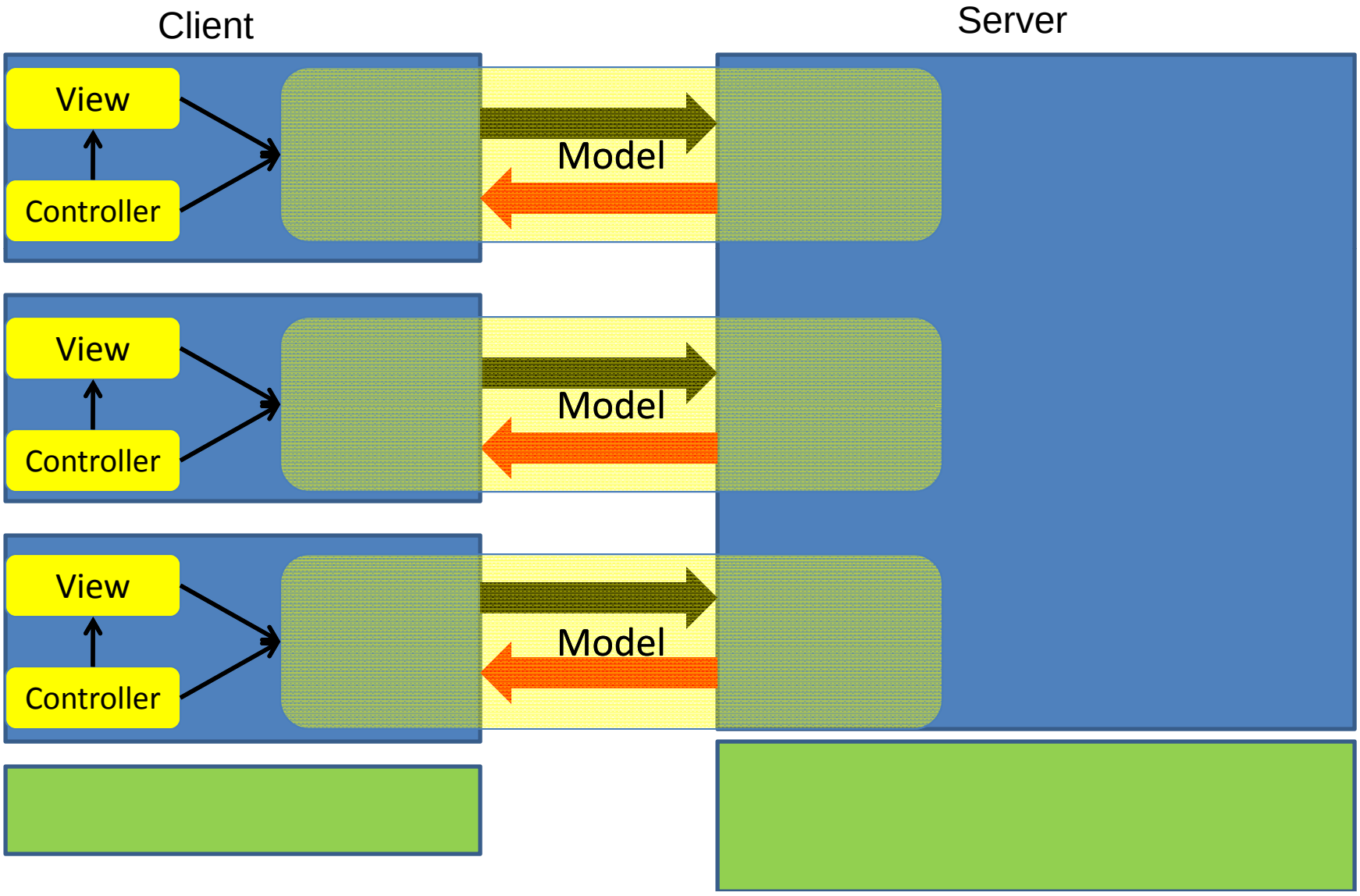
Basic types of structures

- Module
  - Like a type definition
  - Specifies behavior/constraints
- Component/connector
  - Dynamic
  - Exercises the multiplicities
- Deployment
  - Interacts with environment

Styles

- Layered
- Tiered
- Model/view/controller
- Client/server
- Blackboard
- Pipe and filter
- Service oriented
- Enterprise service bus
- …

Integrating styles



Architecture model

- Use an explicit architecture description language rather than a general purpose language like UML
- We used AADL
- It is
  - Standardized
  - Expressive
  - Extensible
  - Supports quality attributes

Error design

error model Example1

features

ErrorFree: **initial error state;**

Failed: **error state;**

Fail, Repair: **error event;**

CorruptedData: **out error propagation**

{Occurrence => fixed 0.8};

end Example1;

error model implementation Example1.basic

transitions

ErrorFree-[Fail]->Failed;

Failed-[**out CorruptedData]->Failed;**

Failed-[Repair]->ErrorFree;

properties

Occurrence => **poisson 1.0e-3 applies to Fault;**

Occurrence => **poisson 1.0e-4 applies to Repair;**

end Example1.basic;

Views and Viewpoints

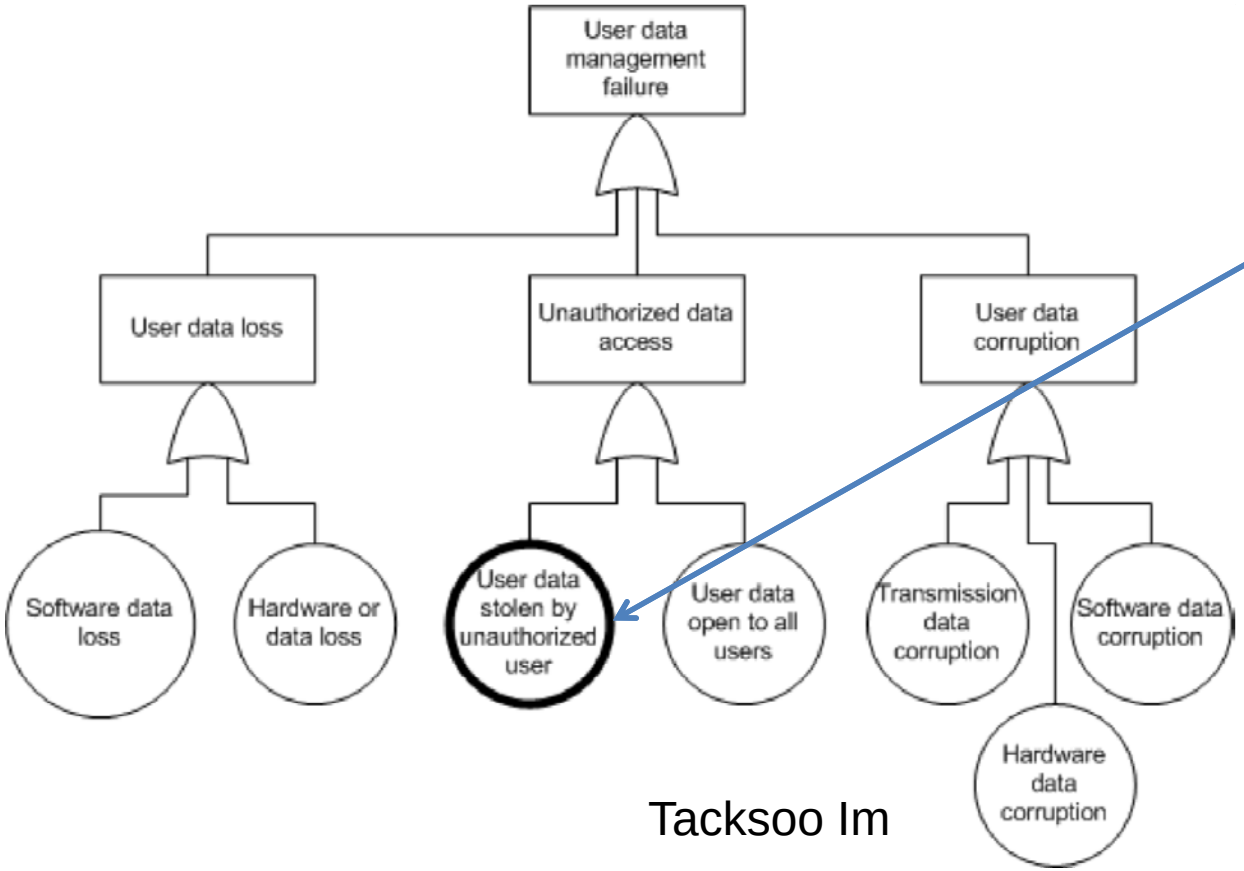
Ocarina

- Petri net shows complexity
- This representation supports simulation

Qualitative Reasoning Framework (cont'd)

Initial Safety Analyses

- FTA (Fault tree analysis) is performed on safety critical hazards identified from the FHA.

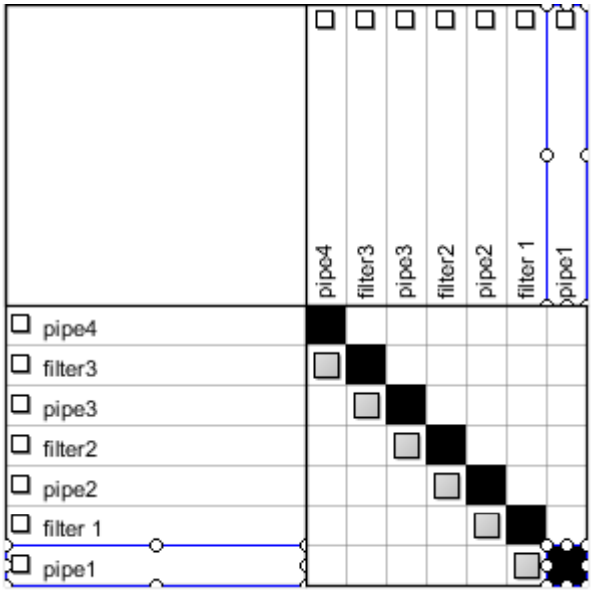


Root cause of the undesired event

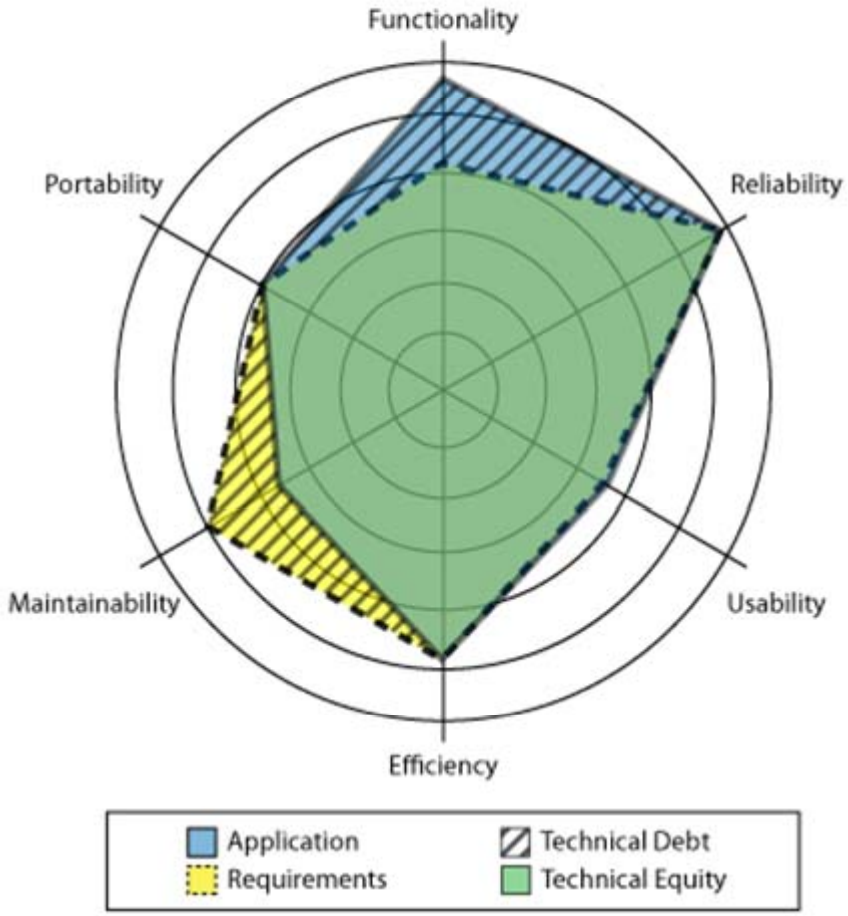
Root causes related to quality attributes are inputs to the reasoning framework

Tacksoo Im

Pipe and Filter DSM



Technical debt is a gap



http://blog.acrowire.com/technical-debt/the-stakeholder-perspective-conclusion/

Total cost of ownership

- Does not stop at delivery of the first version or the nth version.
- On average we spend 80% of total cost on maintenance after first version.
- The part of this that goes to repay technical debt (i.e. fixing things) is a drain on the organization.
- Think what you could do with the money you pay in interest on your credit cards.

The Goal

- Develop software intensive systems that meet the needs of the customer and can be sustained over their intended lifetime.
- The software architecture plays a central role in achieving that goal.
- The software architect plays an important technical and managerial role in a project.

Miss anything?