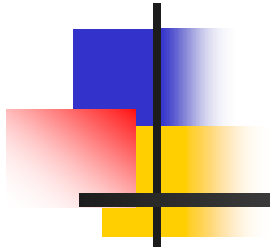


Research Project (Comp8790)



presented

By

Hassan Almari



Presentation Plan

- Presentation Plan
- Project title
- Project Scope
- Project plan
- Patterns brief
- Quality attributes brief
- Relation matrix
- References
- Questions?



Project title

**Investigation
of
the relationship between software (architecture/design)
patterns and quality attributes**



Project Scope

- Identifying and comparing core software architecture and design patterns can be very helpful for software development. Enabling designers and developers to choose the right patterns for the right project or application is a far more effective reusability technique than pure code reuse.



Project Plan

- **Preparation phase**

- History of the patterns and their sources
- Quality attributes how many & what each one mean in software environment
- Start recognition of the relation between QA with the patterns

- **Second Phase**

- Gathering the software patterns and identify which quality attributes they support

- **Analysis phase (most important phase)**

- Analyzing the patterns (chosen patterns) makes relation between patterns styles, patterns fields and patterns relationship with the quality attributes

- **Result phase**

- Building matrix that explain that relations in software context



Patterns brief

- The current use of the term "pattern" is derived from the writings of the architect ***Christopher Alexander*** who has written several books on the topic as it relates to urban planning and building architecture (1964-1979 CA).

Patterns help transporting knowledge and provide common names for solutions.

Pattern Definitions:

- The pattern is, in short, at the same time a thing, which happens in the world, and the rule which tells us how to create that thing, and when we must create it. It is both a process and a thing; both a description of a thing which is alive, and a description of the process which will generate that thing . (p. 247) CA [*TTWoB*].
- Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution. CA [*TTWoB*].



Patterns brief

More Pattern Definitions:

- A "pattern" has been defined as "an idea that has been useful in one practical context and will probably be useful in others."

M.Fowler, "Analysis Patterns – Reusable Object Models, Addison Wesley

- Dirk Riehle and Heinz Zullighoven give a nice definition of the term "pattern" which is very broadly applicable:

A pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.

<http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>



Patterns brief

Kind of patterns in software context:

1. Architectural Patterns

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships between them. The term also refers to documentation of a system's software architecture.

Architectural patterns are high-level strategies

2. Design Patterns

A *design pattern* provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes commonly recurring structure of communicating components that solves a general design problem within a particular context.

Design patterns are medium-scale tactics



Patterns brief

3. Programming Patterns (idioms or coding patterns)

is a low-level pattern specific to a programming language. An idiom describes how to implement particular aspects of components or the relationships between them using the features of the given language. POSA, Volume1 (page 345)

Low-level internal or external details of a component's structure or behavior

4. Conceptual Patterns

A *conceptual pattern* is a pattern whose form is described by means of terms and concepts from an application domain. conceptual patterns focus on system structure in the description of problem and solution. <http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>

There are more kinds of patterns!



Quality attributes Brief

Definition of Quality

One of the challenges of Software Quality is that "everyone feels they understand it.

- **Software quality is the degree to which software possesses a desired combination of attributes (e.g., reliability, interoperability) [IEEE 1061].**
- **A definition in Steve McConnell's *Code Complete* divides software into two pieces: internal and external quality characteristics. External quality characteristics are those parts of a product that face its users, where internal quality characteristics are those that do not**
- **Another definition by Dr. Tom Demarco says "a product's quality is a function of how much it changes the world for the better. This can be interpreted as meaning that user satisfaction is more important than anything in determining software quality.**



Quality attributes Brief

Definition of Quality Attributes

- **Are a characteristics that any system(have/should have) to define the degree of the system quality.**
- quality attributes are non-functional requirements used to evaluate the performance of a system. These are sometimes named "ilities"??

<http://www.vanq.org/Software Quality Attributes.pdf>



Some of Quality Attributes

- **Availability:** Is it available when and where I need to use it?
- **Efficiency:** How few system resources does it consume?
- **Flexibility:** How easy is it to add new capabilities?
- **Installability:** How easy is it to correctly install the product.
- **Integrity:** Does it protect against unauthorized access, data loss?
- **Interoperability:** How easily does it interconnect with other systems?
- **Maintainability:** How easy it is to correct defects or make changes?
- **Portability:** Can it be made to work on other platforms?
- **Reliability:** How long does it run before experiencing a failure?
- **Reusability:** How easily can we use components in other systems?
- **Robustness:** How well does it respond to unanticipated conditions?
- **Safety:** How well does it protect against injury or damage?
- **Testability:** Can I verify that it was implemented correctly?
- **Usability:** How easy is it for people to learn or to use?
- Security & performance

And there are More QAs



Most important for Quality Attributes

- **They are depend on the objectives of the system that will be defined according to the system context and the stakeholders.**
- **Different systems require different attributes to be the most important, for example military systems concern about (Reliability, Security) while the normal commercial systems care more about the Usability .**

Relation matrix

	Quality attributes	Security	Performance	Safety
Patterns name	Patterns Type			
Patterns 1		Scale		
Patterns 2				
Patterns3				

Draft

- I am still in the first phase and the project need a lot of effort to come up with good result?
- I will specify the scale later in the analysing phase.



References

- **The Timeless way of building [TTWoB].(by Christopher Alexander (CA).**
- **A pattern Language by CA.**

- **Design patterns Element of Reusable (GoF) by Erich Gamma,Richard Helm,Ralph johnson,John Vlissides.**

- **Pattern-oriented Software Architecture (POSA Books) by Frank buschmann, Hans rohnert, Regine meunuier,Peter Sommerlad,Michael stal**

- **Software Architecture in Practice by Len Bass,Paul Clements,Rick Kazman.**

- **<http://portal.acm.org/citation.cfm?id=1126207>**
- **<http://www.inf.ed.ac.uk/teaching>.**
- **[http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science)).**
- **http://www.vanq.org/Software_Quality_Attributes.pdf**



Questions?

Thank you