Three Pillars and Five Princes

A Story About Architecture and Design in Software Engineering

Gertjan Zwartjes (Espresso, Sagantec, Code-muse) gertjan@code-muse.com

My Story

- From my experience
- Solving “a problem” with an “automated solution”
- Today, I want to go back to basics: abstractions

A Means to an End

- Common mistake: means *become* and end
- The goal: working product, happy users
- Abstractions are a means, not an end

Part I

The Three Pillars of Abstraction

Abstraction Foundations

- What is abstraction?
- Why do we need abstractions?
- Foundation for reasoning

The Three Pillars

1) Function
2) Architecture
3) Implementation

Pillar I – Functional

- Reason about *what* to build, not *how*
- Terms of function understandable for users too
- Toughest pillar to build

Pillar II – Architectural

- Reason about *plan*, *blueprint*, *design*
- Terms of algorithms, data-structures, building blocks
- Modern stuff: OOP, UML
- Often this pillar is built halfway, or skipped at all

Pillar III – Implementation (1)

- Need I say more?

Pillar III – Implementation (2)

- Yes, maybe I do
- Independent from programming language
- General concepts

No Waterfalls, No BDUF

- Abstractions are independent from software engineering process.
- “Separation of concerns” [E.W. Dijkstra].

Part II

The Five Principles of Software Design

Building Pillar no II

- Gradually more acceptance in the field
- As a *means*
- Why is it important?
- Are there basic elements?

Architecture, But Why?

- Focus on solution, without details of code
- Communication
- Communication
- Communication

Do Not Blindly GSMLAYATFTML!

- Generally, a developer will Grab-Some-Modelling-Language-And-Yet-Another-Tool-For-That-Modelling-Language and start drawing
- What are the basic elements?
- Independent of medium

The Five Princes

- Five principles for software design
- A means to communicate

The Prince of Entity

- Divide system into parts
- Parts are called entities
- Separation of concerns

The Prince of Hierarchy

- Orders entities
- Zoom in and out for more and less detail
- Again separation of concerns

The Prince of Inheritance

- Two additional principles for interrelationships
- First one: inheritance
- Technique for re-use

The Prince of Instances

- Second one: instance relation
- Another technique for re-use

The Prince of Explanation

- design

  = blueprint + behaviour

  = static structure + dynamics
- Explanation brings static structure to life
- The most powerful prince

All The Five Princes Together

1) Entities
2) Hierarchy
3) Inheritance relations
4) Instance relations
5) Explanation

Conclusion

- The Three Pillars
- The Five Princes
- A means
- To a happy end

Thanks for Listening!

For more info: http://code-muse.com