Identifying Design Problems in the Source Code

A Grounded Theory

Thiago Alexandre de Souza Silva

Authors

- PUC-Rio, Rio de Janeiro - RJ
  - Leonardo Sousa, Anderson Oliveira, Willian Oizumi, Simone Barbosa, Alessandro Garcia, Marcos Kalinowski, Rafael de Mello, Roberto Oliveira, Carlos Lucena, Rodrigo Paes

- Lancaster University, Lancaster, Lancashire
  - Jaejoon Lee

- UFAL, Maceio - AL
  - Baldoino Fonseca

Design Problems in Software Systems

- The development and maintenance of **long-lived software systems** require special attention to **non-functional requirements**
  - maintainability, extensibility, availability and performance.

- A **design problem** is the result of one or more **inappropriate** design decisions
  - Impact **negatively** non-functional requirements.

- Design problems were directly related with a **significant increase** in software project **costs.**
  - Leads to the **rejection** of code contributions;
  - Developers should identify them as **early** as possible.

Challenges Identifying Design Problems

- Identifying a single design problem can itself quickly turn into a **very complex** task
  - Documentations are often **unavailable** or **outdated**;
  - Analyze several elements **in source code** to identify each design problem.

- A **single** design **problem** often manifests as **multiple symptoms** scattered in several program elements
  - There is **limited understanding** about how developers identify design problems in practice;

- Solutions for assisting developers in identifying design problems
  - Rely on a **single type** of **symptom** e.g., **code smells** or design principle **violations**

How do developers identify design problems in source code?

Methodology

- Multitrial industrial experiment
  - Professional software developers from five different companies;
  - Identify design problems in their systems under development.
- Captured data on their behaviour
  - Recording audio and capturing their computer screens on video.
- In-depth qualitative analysis based on Grounded Theory procedures.

Objective

- Theory of design problem identification.
  - Insightful propositions and explanations on how design problems are identified;
  - Can serve as a basis to improve the state-of-art.

Concepts and Motivation

- Design Problems Identification
  - Software systems tend to be increasingly large in size and complexity,
  - Each design problem usually pervades the implementation of several elements
    - Developers need to analyze several elements to identify a single design problem.
  - Design documentations are often unavailable or outdated,
    - The source code the only artifact available for developers to identify design problems.

- Grounded Theory
  - Qualitative research method that uses a systematic set of procedures to inductively develop a theory about a phonomenon;
  - Coding procedures (Strauss's and Corbin's GT):
    - Open Coding, Axial Coding and Selective Coding

Research Design

Software Systems and Developer's Selection

- Five Companies from North and Northest of Brazil.
- Specific Softwares with the characteristics:
  - Systems in differents stages of **design degradation**;
  - Systems from different **domains** and **sizes**;
  - Systems there were **not** in their **initial version**;
  - Systems developed in **JAVA**.

Developers were indicated by the companies' managers...

- They were **familiar** with each system and who could act as subjects of the study;
- The subjects were divided into **teams.**

Experimental Tasks

- Activity 1: **Subjects characterization.**
  - Questionnaire about educational level, professional experience, Java programming, and knowledge about design problems.

- Activity 2: **Training.**
  - Training session about software design and design problems, with examples of problems.

- Activity 3: **Problem identification.**
  - A 90 minutes problems design identification in their software systems.

- Activity 4: **Follow-up.**
  - Questionnaire about their perception of the task and indicate whether each symptom was useful to identify a design problem.

Data Collection and Analysis

- All their procedures were **recorded** on **audio and video;**
- Perform **open coding** to associate **codes** with **quotations** of developers' utterances;
- We related the codes through **axial coding**.
  - Codes were **merged** and **grouped** into more **abstract categories;**
- Usage of **selective coding** to identify **core categories** that best explain how developers **identify** design problems.
- **Sjøberg's framework** to **represent** and **describe** the theory **constructs, propositions, explanations and scope**

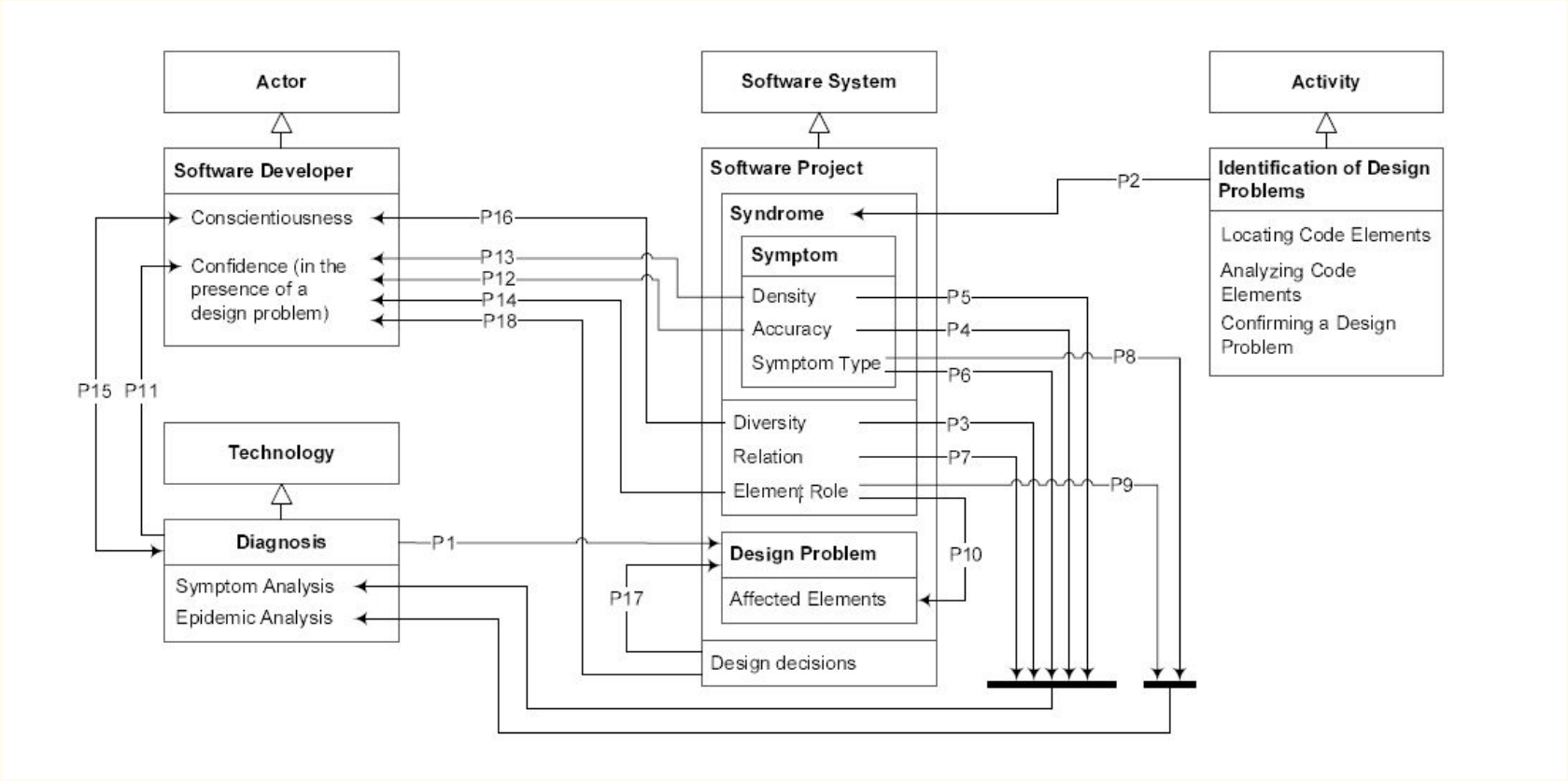
A THEORY ON HOW DEVELOPERS IDENTIFY DESIGN PROBLEMS

GT Definition

- Sjøberg's framework to describe the theory
  - Theory fits in the **Explanation type**
    - **Describes** and **explains** how the identification of design problems is conducted;
    - The **symptoms** and their **characteristics**;
    - How the symptoms are used to **diagnose design problems.**

- Theories evaluation criteria
  - **Testability** - "the degree to which a theory is constructed such that empirical refutation is possible by replicating the study".
    - Experiment conducted with three companies and then replicated it with two more companies to reach theoretical saturation.

Constructs and Propositions



Identification of Design Problems

Constructs

- C1 - **Design Problem** - A design decision that negatively impacts quality attributes;
- C2 - **Design Decisions** - Decisions made during the software development process;
- C3 - **Symptom** - An indication of the presence of a design problem;
- C4 - **Syndrome** - A set of symptoms affecting the same code element;
- C5 - **Diagnosis** - The process of identifying a design problem through the analysis of symptoms that manifest themselves in the source code.

Propositions

- P1 - The diagnosis affects the identification of design problems;
- P2 - Identification of design problem has three steps in which developers relies on design problems symptoms in all of them;

Design Problem Symptoms

Constructs

- C6 - **Symptom Type** - A category to which a set of symptoms with common characteristics belongs;
- C7 - **Accuracy** - The degree to which a symptom is correct in indicating a design problem;
- C8 - **Density** - The number of symptoms instances in a syndrome;
- C9 - **Relation** - How two or more symptoms are connected;
- C10 - **Diversity** - The degree to which a syndrome contains a variety of symptom types.

Propositions

- P3 - The diversity of symptoms influences which symptoms the developer will use during the diagnosis

Design Problem Diagnosis

Constructs

- C11 - **Symptom Analysis** - The process of analyzing a set of symptoms affecting a single element;
- C12 - **Epidemic Analysis** - The process of analyzing elements affected by the same set of symptoms.

Propositions

- P4 - The symptom accuracy influences which symptoms the developer will use during the diagnosis;
- P5 - The density of symptoms influences which symptoms the developer will use during the diagnosis;
- P6 - The type of symptom influences which symptoms the developer will use during the diagnosis;
- P7 - The relation among the symptoms influences which symptoms the developer will use during the diagnosis;

Design Problem Diagnosis

Constructs

- C11 - **Symptom Analysis** - The process of analyzing a set of symptoms affecting a single element;
- C12 - **Epidemic Analysis** - The process of analyzing elements affected by the same set of symptoms.

Propositions

- P8 - The type of symptom affects the developer's choice of a epidemic element to be analyzed;
- P9 - The role that the element plays on the system affects the developer's choice of a epidemic element to be analyzed;
- P10 - The role that the element plays on the system is associate with the confirmation of the presence of a design problem.

Propositions Concerning the Developer

Confidence in the Presence of a Design Problem

Constructs

- C14 - **Confidence** - The degree to which they are convinced about the presence of a design problem.

Propositions

- P11 -The diagnosis affects the developer's confidence in the presence of a design problem;
- P12 - The symptom accuracy affects the developer’s confidence in the presence of a design problem;

Propositions Concerning the Developer

Confidence in the Presence of a Design Problem

Constructs

- C14 - **Confidence** - The degree to which they are convinced about the presence of a design problem.

Propositions

- P13 - The density of symptoms affects the developer’s confidence regarding the presence of a design problem;
- P14 - The role that the element plays on the system influences the developer's confidence

Propositions Concerning the Developer

Conscientiousness

Constructs

- C15 - **Conscientiousness** - A personality trait related to being careful, responsible, and persevering.

Propositions

- P15 - The consciousness affects the likelihood of a developer identify a design problem;
- P16 - The diversity of symptoms affects the conscientiousness of the developers.

Propositions Concerning the Developer

Incapability of Providing an Alternative

Constructs

Propositions

- P17 - The design decisions affects the confirmation of the presence of a design problem;
- P18 - The design decisions affects the confidence developer’s confidence in the presence of a design problem.

Towards Improving Design Problem Diagnosis

Future Works

- Supporting Multiple Symptoms
  - **Automatic** tool could help developers to **filter** several **symptoms** that are most **likely** to indicate a **design problem.**
  - Tool to provide **visualization** mechanisms to support.
- Prioritization of Similar Elements
  - Researchers could use the attributes presented to build tools that **prioritize role elements**.
- Additional Support for the Developer
  - Tool to indicate an **alternative implementation** that could **remove** the design problem **symptoms**;
  - Tools that allow developers to **personalize** the **detection** of symptoms according to their software systems

Threats to Validity

- Construct Validity:
  - Symptoms provided for developers could have biased the experiments;
  - Data provided considering the literature.
- Internal Validity
  - Developers' background knowledge can be a threat;
  - Provided training to mitigate this threat.
- External Validity
  - The number of subject represents a threat;
  - Is a multi-company study involving different working environments and systems.
- Conclusion Validity
  - The participation of the author who followed the GT procedures;
  - The GT coding activities were shared with other researchers.

Conclusion

Conclusion

- Derived theory describing the **activities** and **factors** that influence on **how developers identify design problems**
  - Further understand the **identification** of **design problems**.
- The theory reveals that developers rely on a **heterogeneous** set of **symptoms**, and they tend to **combine** symptoms.
- The theory also presents the **characteristics** of **symptoms** that developers consider **helpful.**
- Theory can be used to **advance** the **state-of-art.**
- **New** empirical studies to **assess** in more **depth** the theory's **propositions** and **explanations.**

Thank You!

Doubts?