Developing Measurements for Software Architectural Smells in the Domain of Enterprise Architecture Debts

Benny Tieu
A pre-study to Master Thesis at EECS, KTH

Developing Measurements for Software Architectural Smells in the Domain of Enterprise Architecture Debts

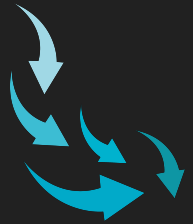
Benny Tieu
A pre-study to Master Thesis at EECS, KTH

Technical Debt (TD)

Technical Debt a term for parts in software development that, over time, causes...

...higher costs

and lower quality

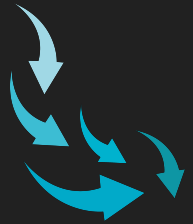


Technical Debt (TD)

Technical Debt a term for parts in software development that, over time, causes...

...higher costs

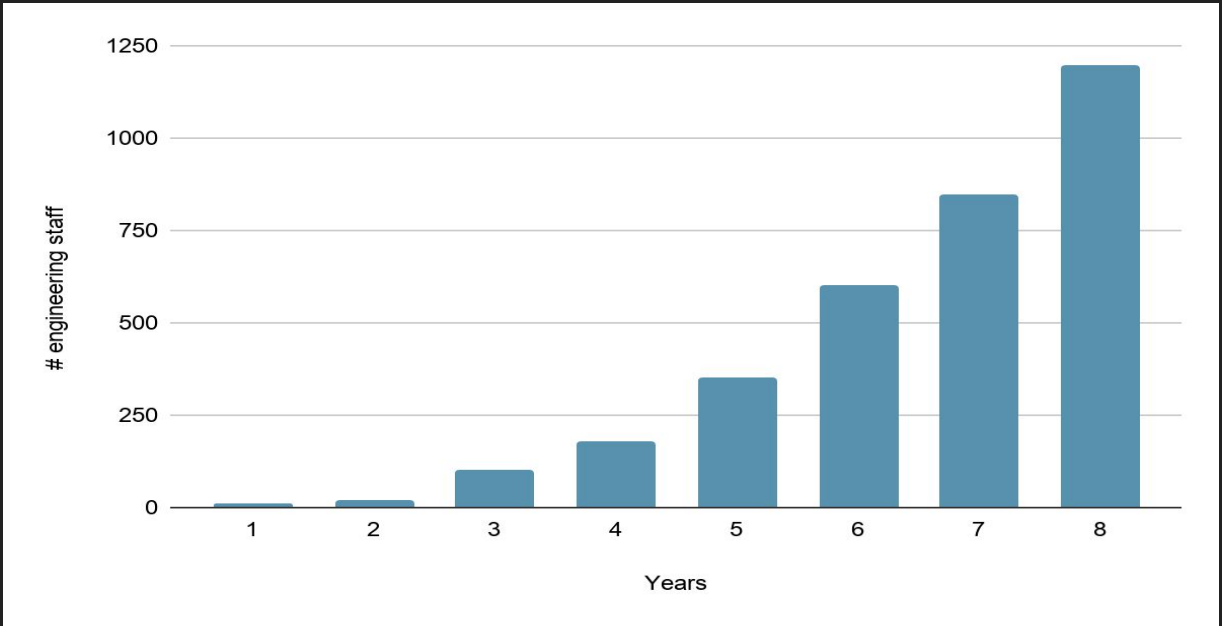
and lower quality



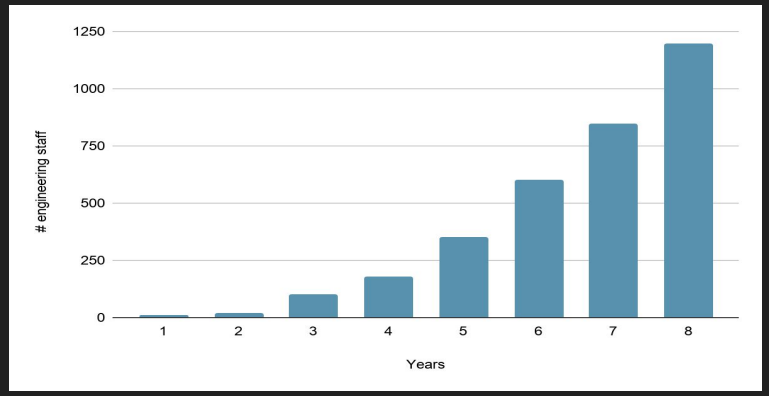
Technical Debt (TD)

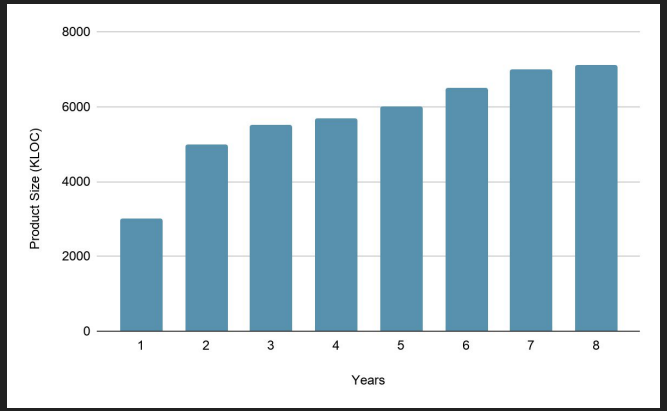
Example of a market leading software company

- Successful software?

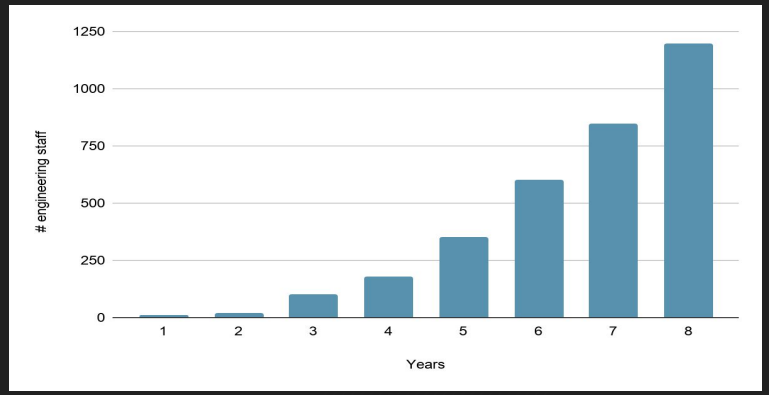


Technical Debt (TD)

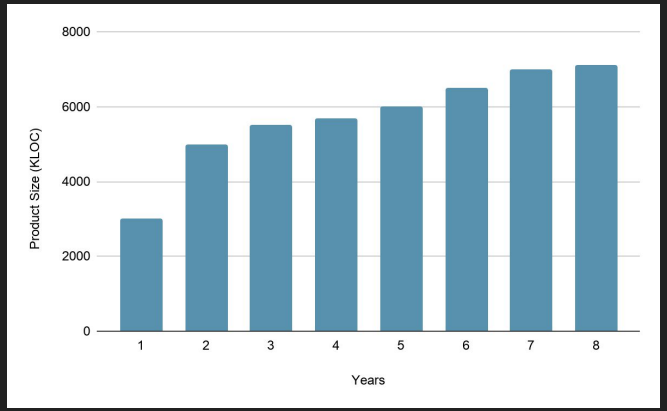




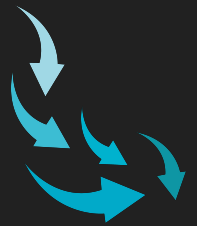
Technical Debt (TD)



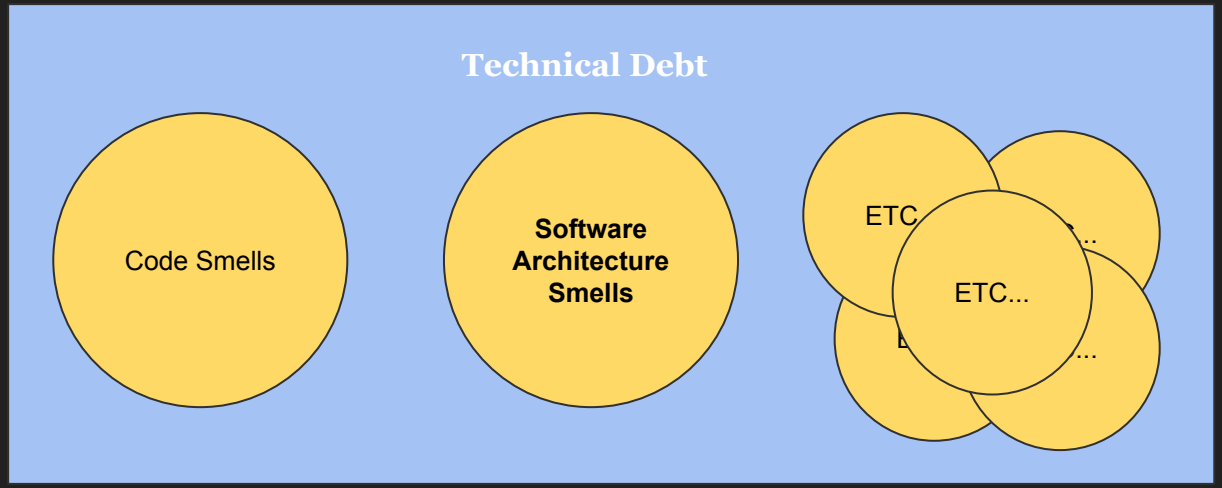
...higher costs



and lower quality

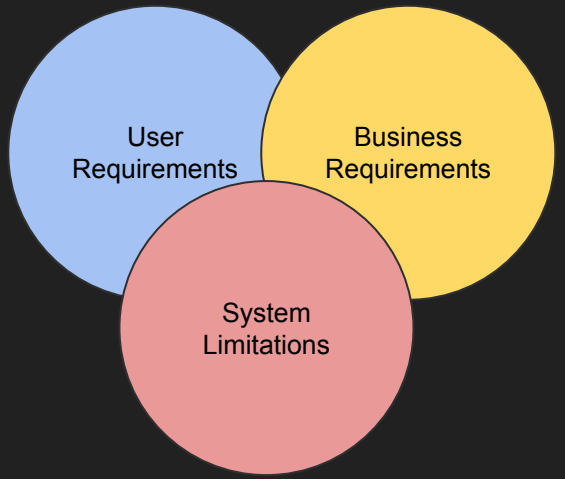


Technical Debt (TD)



Software Architecture (SA)

- Structure of a software system
- How components interacts with each other



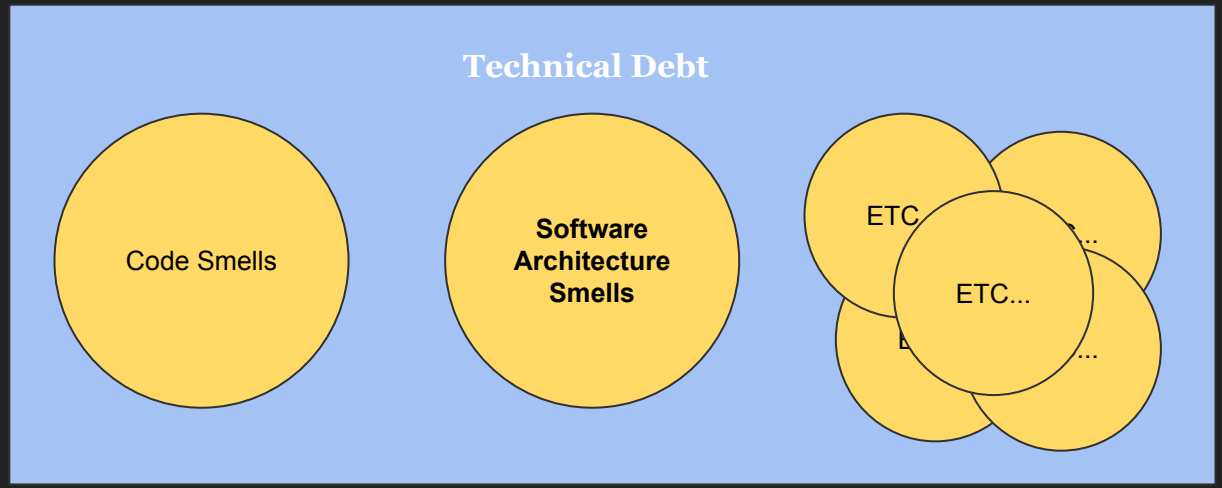
Software Architectural Smell (SA)

Good SA: “**Minimize** the **human resources** required to **build and maintain** a **software system” (Martin 2019)**

- If effort is increased and the productivity decreased => Bad SA
- If the effort is low and stays low throughout the system lifetime => Good SA

- Symptoms of bad behavior or anti-pattern is called a **SA Smell**

Limitations of Technical Debts



Enterprise Architecture Debts and anti-patterns

Enterprise Architecture Debt (EAD)

From “Towards a definition of enterprise architecture debt” (Hacks et al):

“Enterprise Architecture Debt is a metric that depicts the deviation of the currently present state of an enterprise from a hypothetical ideal state.”

“Towards a Catalog of Enterprise Architecture Smells” (Salentin & Hacks)

- Transferred concepts of **code smells to the domain of EAD**
- The transferred concepts result will be newly defined EA Smells
- Using an EA model and a software to detect EA Smells

Software Architectural Smells → EA Smells

My thesis will be a extension of "Towards a Catalog of Enterprise Architecture Smells" with the problem statement:

"What corresponding EA smells can be defined on the existing software architecture smells and anti-patterns?"

Examination Method

- Thesis will follow the methods of **Design Science Research** (Peffers et al 2006)

1) ~~Identify the problem and motivate~~
2) ~~Define an objective for a solution~~
3) (Currently here) **Design and development of an artifact**
4) Demonstration by using the artifact to solve the problem
5) Evaluating the solution
6) Communicate the search to researchers or practicing professionals

- Identify known SA smells in order to map them to the domain of EA smells
- The outcome of the development are the EA smells, i.e the artifact

Literature review. First steps

- **Clean Architecture** (Robert C. Martin)
  - A credible author that has written many books about software architecture
  - S.O.L.I.D a well-known set of principles of how to design a good software architecture
  - Breaking the principles = smells

- **Automatic Detection of Instability Architectural Smells** (Fontana et al)
  - Well cited paper
  - A software to automatically detect software architectural smells

Questions and limitations

- Are there any overlapping in between code smells and SA smells?
  - What are their relation to each other?

- There is a lot of research and paper in SA smells, how to know where to limit and know what is most credible?

- The validity of the EA model may be a limitation.

To be continued...

Thanks!