Understanding the development of data analytics solutions in a small development team: A case study

Alex Bruno Paranhos da Silva
Daniel Amador dos Santos

Agenda

1. Introduction
2. Design
    a. Problem
    b. RQ’s
    c. Case selection
    d. Procedures and roles
    e. Data collection
    f. Analysis procedure
3. Results
    a. Inductive software engineering principles
    b. Non-functional requirements
4. Conclusion
5. References

INTRODUCTION

Background

- Data Analytics
- Software Analytics
- Inductive Software Engineering manifesto (Menzies et al. 2011).

Background

- Principles for Software Analytics
  - Users before algorithms
  - Plan for scale
  - Early feedback
  - Be open-minded
  - Do smart learning
  - Live with the data you have
  - Broad skill set, big toolkit
- Not a formal maturity model
- Good practices that are valuable
- Following this principles leads to good data analytics solutions (Menzies & Zimmerman, 2013).

Background

- Non-functional requirements.
- Related to quality → execution time (Mari and Eila, 2003)
  - Performance
  - Security
  - Availability
  - Usability
  - Scalability
  - Reliability
  - Interoperability
  - Adaptability

Problem

What are the practices used to develop data analytics solutions by a small software team concerning data mining procedures and non-functional requirements?

DESIGN

RQ’s

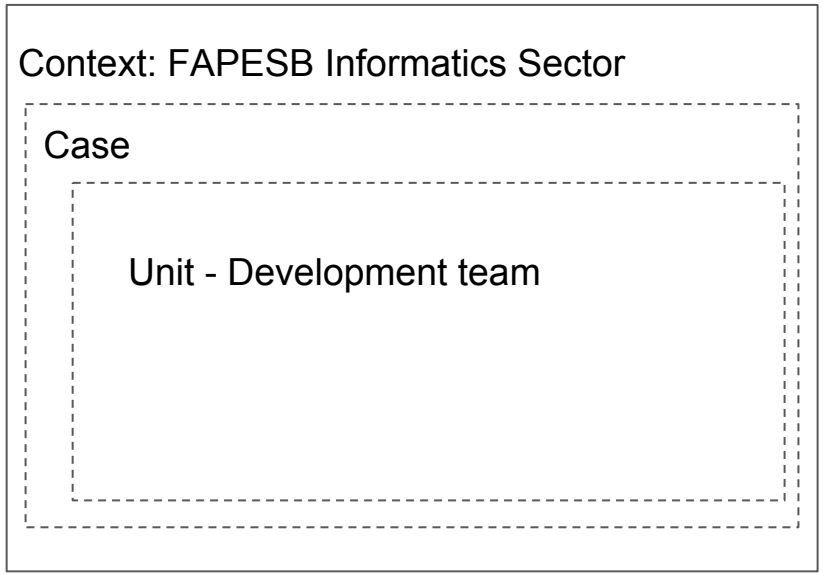
- **RQ1**. Are Inductive Software Engineering Principles followed by the development team?
- **RQ2**. What are the non-functional requirements used in the development of data analytics solutions?
  - **RQ2.1.** How these non-functional requirements are specified, designed, implemented and tested?

Case Selection

- FAPESB
  - Informatics Sector
    - Infrastructure team
    - **Software Development team**
- Observatório and Sistemas de Bolsas

Design

- Embedded case

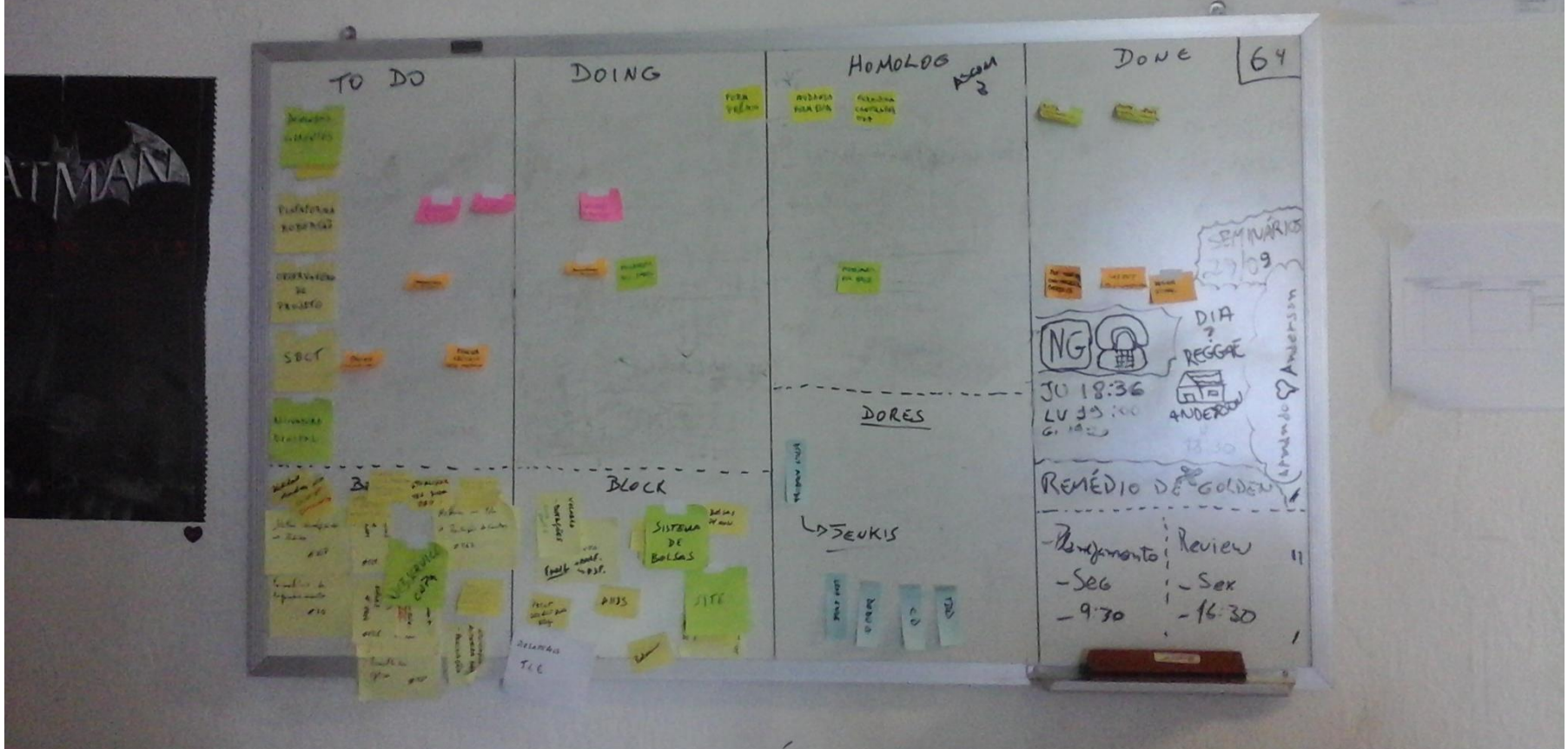


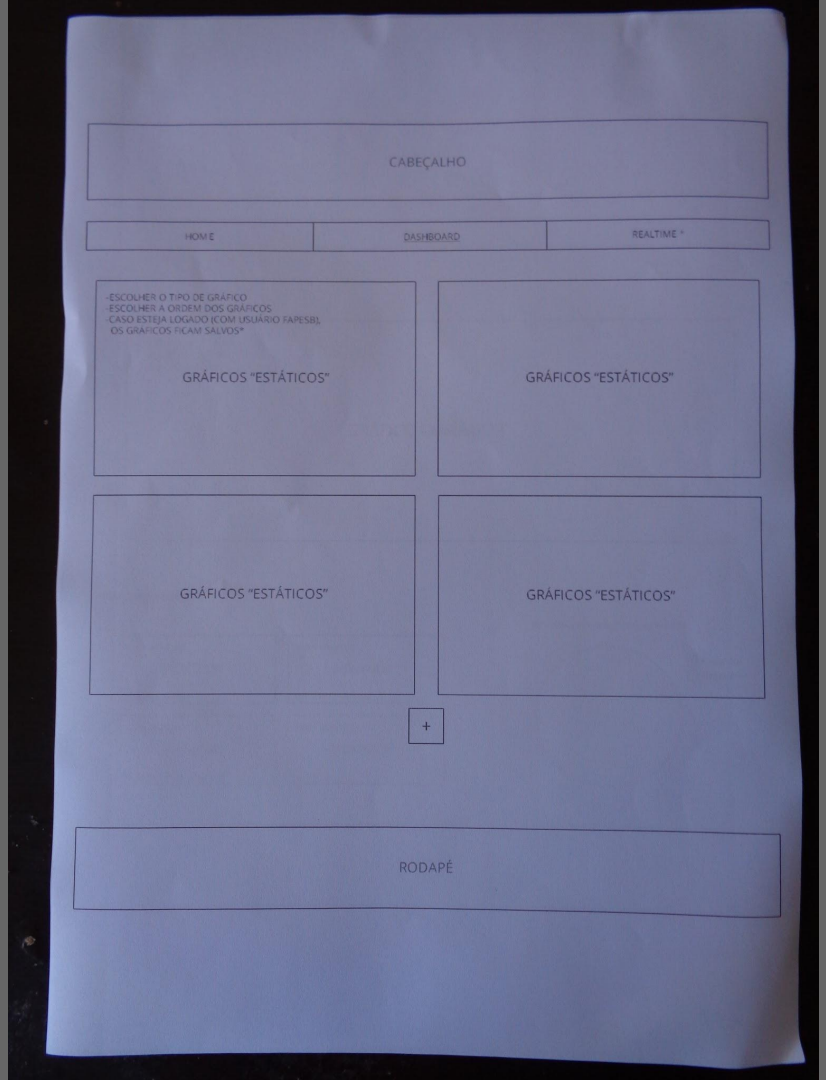
Procedures and roles

- Interviews with the development team
  - Semi-structured session
  - Funnel general principle
  - Each person at time
- Pilot session
  - 1 current developer
- One researcher conduct the interview and the other take notes and assists the session
- 4 participants
  - 2 current developers (D1 and D2)
  - 1 former developer (F1)
  - 1 stakeholder/statistician (S1)

Data collected

- Audio collected from the interviews (with further transcribing)
- Developers’ personal notes
- Graphic User Interfaces models
- Pictures from the workplace
    - Whiteboard


TO DO
DOING
HOMOLOG
DONE
64
SEMINÁRIO
DIA ? REGGAE
NG
ANDERSON
DORES
BLOCK
SISTEMA DE BOLSAS
SITE
REMÉDIO DE GOLDEN
Review
- Sex
- 16:30
- 9:30


CABEÇALHO
HOME
DASHBOARD
REALTIME *
-ESCOLHER O TIPO DE GRÁFICO
-ESCOLHER A ORDEM DOS GRÁFICOS
-CASO ESTEJA LOGADO (COM USUÁRIO FAPESB),
OS GRÁFICOS FICAM SALVOS*
GRÁFICOS "ESTÁTICOS"
GRÁFICOS "ESTÁTICOS"
GRÁFICOS "ESTÁTICOS"
GRÁFICOS "ESTÁTICOS"
+
RODAPÉ

11/04/16 (S47)

~~Seg~~ Ter Qua Qui Sex Sáb Dom

Ponto de Março
- ↳ Casos da FAFA (EMAIL)
- ↳ Ñ consigo abonar o dia 24 de Jul.
- ↳ Juliana à partir de 14/03

~~Observatório~~ OK
- ↳ ~~Municípios [illegible]~~ OK
- ↳ ~~Municípios Atualizados~~ OK

~~Portal → Plugin Share~~ OK

~~Siga PRO~~
- ↳ ~~Atualizar Tabela de área (Tecnologia)~~
  - → ~~Ver formulário Ativos~~

~~Foto c/ Teporb - Admin~~
- → ~~Foto do Assunto~~

Sistema não funciona S/ www
- ↳ ~~EMAIL ⇒ Aguardando Resposta~~

~~Criar Texto Facilidades~~

~~C[illegible]~~
- ↳ ~~ASSinatura (Tempo)~~
- ↳ [redacted] VooR!

~~Ver com Lorena a Impressão dos Relatórios c/ programas (Dígito Segurança)~~

~~Ver Programas na base e nas telas~~

S/A ⇒ ID ✓ ⇒ "Programas Educação" OK

FORONI

Legislação → Falta "S" OK

Artifacts *in loco*

- Java Source Code
- Databases schemas
- Observatório database diagram
- Observatório project structure (Java + annotations + JavaScript)

Analysis Procedures

From the participants' statements, identify:

- Overall SE process
- Description about principles
- Description about non-functional requirements

→ Triangulation with other artifacts.

RESULTS

Software Engineering

- Team follows a Scrum-like approach
  - Whiteboard
  - Daily meetings
- F1 worked by himself in Observatório project
- Self-managed teams
- Team does not appreciate to have lots of documentation

Users Before Algorithms

- Principle applied.
- D1: “All [the process] is done [thinking] on the final user: even how (s)he will interpret some datum we are exhibiting.”
- F1 : “We went after the data the client wanted to see.”
- S1 had freedom to ask for changes whenever she felt the system was not fulfilling her needs.
- S2: Flow client → designer → programmer

Plan for scale

- Principle partially applied.
- F1: “The algorithm does not discover information by its own”.
- D2 says it is easy to insert new graphics for new indicators.
- The new well-structured database might help mining.

Early feedback

- Principle applied.
- Access to S1 was direct and informal. All subjects agree on that.
- F1 stated that it would be unfeasible to receive feedback only in the latter development stages.

Be Open-Minded

- Principle not applied.
- F1 says that the lack of time prevented him to have an “open-minded” approach.
- S2: “In Observatório, I never saw it happening.”

Do smart learning

- Principle applied.
- All the subjects except for F1 felt the data were validated.
- S1 declares she made statistical validation.

Live with the data you have

- Principle applied.
- It was applied by the sheer absence of an alternative source of data.
- D1: “You have to go with what you have”.
- Researcher visualized databases.

Broad skill set, big toolkit

- Principle not applied.
- Data only about the past.
- No time to change what is done.
- No specific tool for data discovery.

Performance

- Requirement is present
- Motivation from the team
- Actions to improve performance
  - Create new database;
  - Flat design;
  - Relocate the process to the client side;
  - Choose of technology;
  - Design of their architecture.

Security

- Least critical requirement
- According to F1:
  - Read-only system;
  - Public data;
  - Does not store personal information.
- Requirement delegated to infrastructure
  - SQL Injection and DDoS attack

Availability

- Attribute essential to the client
  - External access
- Development team is not in charge of the infrastructure
- Not concern to the team
- According to S1
  - Attribute achieved
  - Never crash or offline
- Did not perform tests

Usability

- Requirement to worry
  - External access
- According to D1
  - Concerns from the team;
  - Not critical to the client;
  - Functionality > Usability.
- Designer
  - Human-computer interaction specialist;
  - Use of prototyping.
- Did not execute tests

Scalability

- Attribute achieved
- According to D1
  - System planned to expand;
  - Requirement from the client;
  - New features and module.
- The team actions
  - MVC pattern;
  - Annotations in Java.
- F1 stated, “the architecture is the key point of system’s scalability”

Reliability

- Reliability issues is not the team’s responsibility
- Experienced a few problems
  - Nothing related to reliability
  - No plan to handle
  - D2 stated, “fix the bugs whenever the developers perceive that something is not working as it should.”

Interoperability

- According to all subjects, is not a concern
  - System does not communicate with other application;
- Migration of data from external database
  - Manually;
  - Once per year.
- Observatório use own database

Adaptability

- Required from the client
  - Desktops;
  - Tablets;
  - Smartphones.
- Use of responsive layout
- Does not execute test formally

CONCLUSIONS

Conclusions

- Principles: 4.5 from 7.
- Non-functional requirements: 4 from 8 managed by the team.
- No development process formalized
- A “detached developer” approach.
- The diminute size of the team brought synergy.
- Applying of the principles depended on the experience and vision from the developers.
- Data discovery and software engineering embedded together.

Threats to Validity

- With more artifacts, the triangulation would be stronger.
- Even with the pilot and with concepts explanation, sometimes the participants could not understand the questions' ideas.
- Participants may have had "disobeyed" us and commented each others the answers before all the interviews were finished.

Schedule

Task	Start Date	End Date	Progress
Write the first version of the protocol	4-Aug-2016	15-Sep-2016	Done
Define the final interview script	1-Sep-2016	12-Sep-2016	Done
Execute the pilot interview	28-Sep-2016	28-Sep-2016	Done
Execute the interviews and collect data	17-Oct-2016	22-Oct-2016	Done
Transcribe interviews	17-Oct-2016	22-Oct-2016	Done
Analyze the data	20-Oct-2016	25-Oct-2016	Done
Reporting	19-Oct-2016	27-Oct-2016	Done
Present	27-Oct-2016	27-Oct-2016	In Progress

References

- Buse, Raymond PL, and Thomas Zimmermann. "Information needs for software development analytics." *Proceedings of the 34th international conference on software engineering*. IEEE Press, 2012.
- Chen, Hsinchun, Roger HL Chiang, and Veda C. Storey. "Business Intelligence and Analytics: From Big Data to Big Impact." *MIS quarterly* 36.4 (2012): 1165-1188.
- Davenport, Thomas H., Jeanne G. Harris, and Robert Morison. *Analytics at work: Smarter decisions, better results*. Harvard Business Press, 2010.
- Ian Sommerville. Software Engineering . Addison-Wesley Publishing Company, USA, 9th edition, 2010.

References

- Mari, Matinlassi, and Niemelä Eila. "The impact of maintainability on component-based software systems." *Euromicro Conference, 2003. Proceedings. 29th*. IEEE, 2003.
- Menzies, T. and Zimmermann, T., 2013. Software analytics: so what?. IEEE Software, 30(4), pp.31-37.
- Menzies, T., Bird, C., Zimmermann, T., Schulte, W. and Kocaganeli, E., 2011, November. The inductive software engineering manifesto: principles for industrial data mining. In Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering (pp. 19-26). ACM.
- Zhang, D., Han, S., Dang, Y., Lou, J.G., Zhang, H. and Xie, T., 2013. Software analytics in practice. IEEE software, 30(5), pp.30-37.