Introduction to the Course
The GQM Model

Why this course?

- To provide a solid foundations for Data Scientists and Software Engineers …
- … for the development of their systems,
- … especially with reference to quantitatively :
  - estimating,
  - planning,
  - reviewing,
  - budgeting.

And also provide the framework for a solid review of fundamental statistical concepts...

Main technological trends

Based on the opinions of strategic investors:

- Multicore Architectures and High Performance Computing,
- Artificial Intelligence, Machine Learning, Big Data technologies, and Data Analytics,
- Next Generation Internet and Internet of Things.

Issues in Data Science (1/4)

Clear need of Software Engineering practices, as evidenced by:

- a large body of literature (Abdellatif et al. (2015)[†]),
- specific high profile workshops on this subject (e.g., BIGDSE)

[†] *T. M. Abdellatif, L. F. Capretz, and D. Ho. Software analytics to software practice: A systematic literature review. In 2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering, pages 30?36, May 2015.*

Issues in Data Science (2/4)

Four major areas, (Madhavji et al (2015)[†]):

1. quantify the effort for the production of very time sensitive systems,
2. requirements, being closer to customer needs and produced very fast, always faster,
3. overall architectures , handling the multiple possible solutions Mistrik et al. (2017)[††],
4. testing and maintaining the resulting systems – two previous lines indeed result in more complex systems to test.

[†] *N.H. Madhavji, A. Miranskyy, and K. Kontogiannis. Big picture of big data software engineering: With example research challenges. First International Workshop on BIG Data Software Engineering, BIGDSE '15*

[††] *I. Mistrik, R. Bahsoon, N. Ali, M. Heisel, and B. Maxim. Software Architecture for Big Data and the Cloud. Morgan Kaufmann Publishers Inc., 2017.*

Issues in Data Science (3/4)

There is a need for new, effective, agile, and flexible organizations†:

- shorter and very predictable lifecycles,
- higher flexibility and adaptability in design of software, coming from the shift from traditional SQL-based to noSQL structures:
  - availability,
  - pervasive distribution,
  - management of heavy and/or workload,
  - intensive Data Analytics;

Issues in Data Science (4/4)

And also:

- new processes and tools supporting:
  - more effective prediction of the time to develop,
  - better testing of systems,
  - more decentralized and agile processes,
  - suitable support of prediction mechanisms to anticipate and evaluate the effort and quality,

[†] *See, e.g., the special issue of IEEE Software by Gorton et al. (2016): I. Gorton, A. B. Bener, and A. Mockus. Software engineering for big data systems. IEEE Software, 33(2), Mar 2016.*

Specific issues in ML (1/2)

ML systems are complex (software) engineering endeavours; just consider the definitions of:

- type of network (fully-connected, convolutional, or recurrent network),
- type of connections (dense, sparse, or skip connections),
- number of layers,
- number of units per layer,
- type of loss function,
- ...

Specific issues in ML (2/2)

The recent developments in ML evidence a clear need of software engineering practices:

- major corporations, like Google and Microsoft are supplementing their tools with suitable software engineering infrastructures:
  - Tensorflow (Seide and Agarwal, 2016[†]);
  - CNTK (Xia et al., 2017[††]).
- the workshop “Software Engineering for Machine Learning” at NIPS,
- “International Workshop on Realizing AI Synergies in Software Engineering” (RAISE),
- specific degrees and certificates like at SUNY Stony Brooks,

[†] *Frank Seide and Amit Agarwal. Cntk: Microsoft?s open-source deep-learning toolkit. KDD ’16*

[††] *C. Xia, · · · . Tensorflow estimators: Managing simplicity vs. flexibility in high-level machine learning frameworks. KDD ’17*

Data Science for SE (1/3)

Data Science can also significantly improve SE practices, since there is evidence that[†]:

- in general software engineers do not have specific skills in Data Science so they need tools (e.g., “software analytic”),
- SE practices can be be improved by the analysis of available software engineering data using software analytics, in particular wrt:
  - internal vs. external analytics,
  - quantitative vs. qualitative methods,
  - data mining vs. interactive tools,
  - target audience,
  - exploratory vs. deployment analytics;
- a portion of this, which now is gaining a self-standing role, is the whole issue of blockchain technologies

[†] *R.P.L. Buse and T. Zimmermann. Information Needs for Software Development Analytics. ICSE ’12.*

Data Science for SE (2/3)

Software Engineering can also significantly benefit from the new work in AI and ML:

- For more than 30 years there have been scientists working on the application of AI and ML to Software Engineering:
  - Khoshgoftaar in the '90,
  - Pedrycz, Succi, and collaborators from 1996,
  - Menzies from the year 2000.
- Most of this effort has spanned the area of monitoring and prediction of software quality and productivity and of generating code and tests.
- So far the results have not yet had a groundbreaking effect.

Data Science for SE (3/3)

- However, the most recent discoveries in Machine Learning may produce a significant move ahead in this area, especially in the areas of:
  - prediction and modelling of the quality, the reliability, and the time and effort to develop software systems, especially taking advantage of deep learning
  - requirement analysis
  - general backoffice tasks, determining if AI/ML algorithms could be more effective than professionals
- We are currently experimenting ML tools to identify skills of developers, reliability of software systems, quality attribues, ...
- In these cases, the results appear particularly promising.

Needs vs. Research in SE

1. To better understand where the software industry should go in general we have performed an empirical study.
2. We have asked the key problems software companies face.
3. We have then compared the result with the most active research areas in software engineering.
4. On one side, we have administered a questionnaire to a sample of developers coming from more than 40 software companies to determine their major concerns in software development,
5. On the other, we have analysed the research outcome of the last major conferences in Software Engineering and we have checked if needs expressed by the developers were matched by the research effort of the scientific community.

Needs in SE

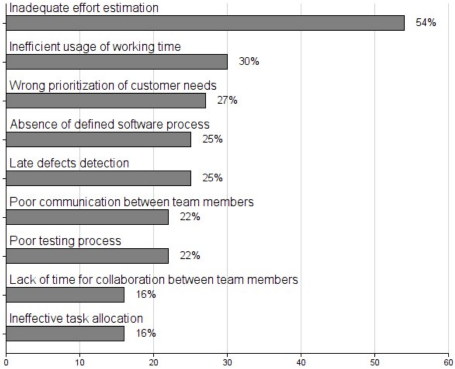
In our questionnaire, we have asked:

1. the biggest obstacles that affect the ability to deliver software,
2. their aimed internal improvements,
3. their planned innovation.

Biggest obstacles in SE (1/2)

- More than half of our survey participants said inadequate estimating efforts is one of the major obstacle.
- Other major obstacles include:
  - inefficient usage of working time,
  - wrong prioritization of customer needs,
  - absence of defined software process, and
  - late defects detection.

Biggest obstacles in SE (2/2)

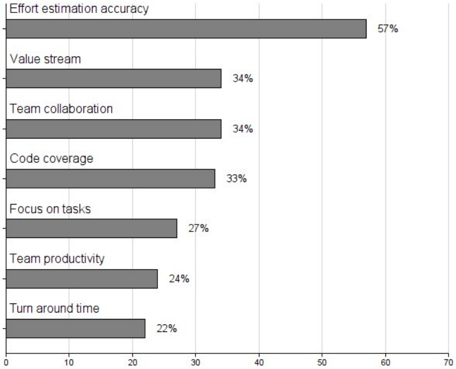


What are the three biggest *obstacles*
that affect your ability to deliver software?

Sought internal improvements in SE (1/2)

- This second question asks the participants to identify what they want to improve in the next project.
- Still, our survey participants showed the keenest interest in improving effort estimation — 57% of the participants answered they want to improve effort estimation.
- It is noteworthy that difficulties in controlling software quality seem to be less of a concern to software engineers.

Sought internal improvements in SE (2/2)

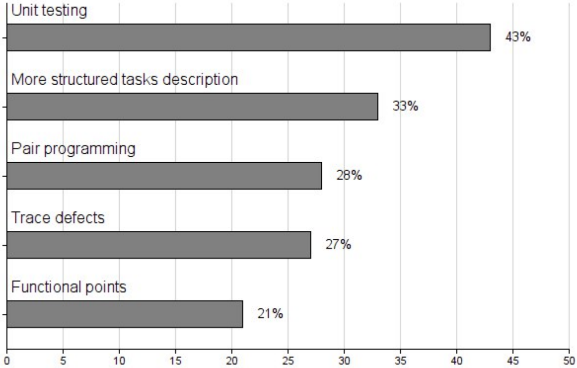


Which three internal *improvements* for your team will you seek for the next project?

Planned innovations in SE (1/2)

- This question helps us understand whether the survey participants are aware of methods/techniques they can use to tackle the difficulties they are experiencing.
- Participants chose *unit testing* most frequently.
- The answers to this question are a bit contradictory.
  - It seemed that the respondent were more interested in code quality than in effort estimation.
- A likely interpretation is that they were not aware of any structured technique for effort estimation
  - since most of the companies claimed to follow an agile process they relied mostly on “agile common sense:”
  - unit testing
  - better task descriptions.

Planned innovations in SE (2/2)

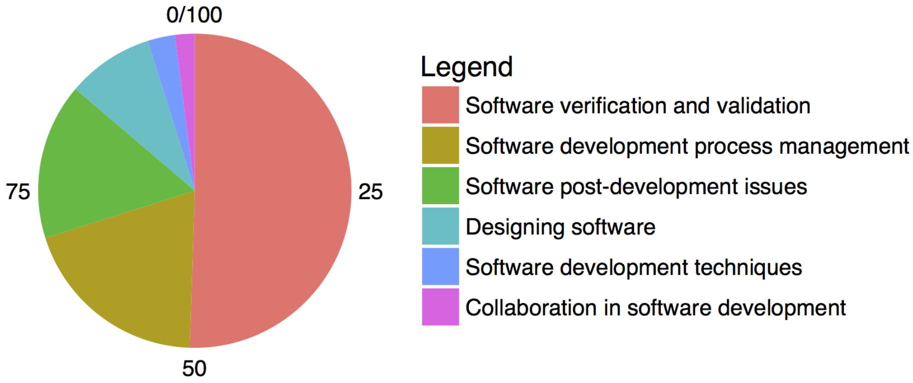


Which three *innovations*
will you introduce to your software process for the next project?

Research in SE (1/2)

- The research papers in ICSE and FSE are mostly in software verification and validation – more than half of the total.
- Publications in the categories of software development process management and software post-development issues account for 20% and 16%, respectively.
- In contrast, little attention is placed to designing software (9%), software development techniques (3%), and collaboration in software development (2%).

Research in SE (2/2)

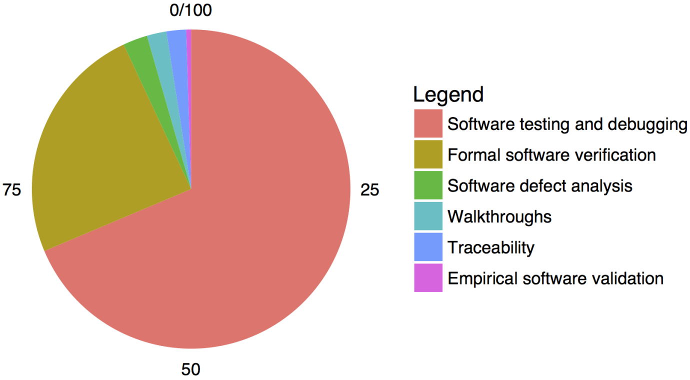


The distribution of ICSE/FSE papers in 2014, 2015, and 2016.

Focus on V & V (1/2)

- Also inside software verification and validation we have found also there a large imbalance
- The more than 80% of papers are published in two very closely related fields, viz., software testing and debugging (69%) and formal software verification (24%).
- More practical research topics like walkthroughs, traceability, and empirical validations score altogether less than 5%.

Focus on V & V (2/2)

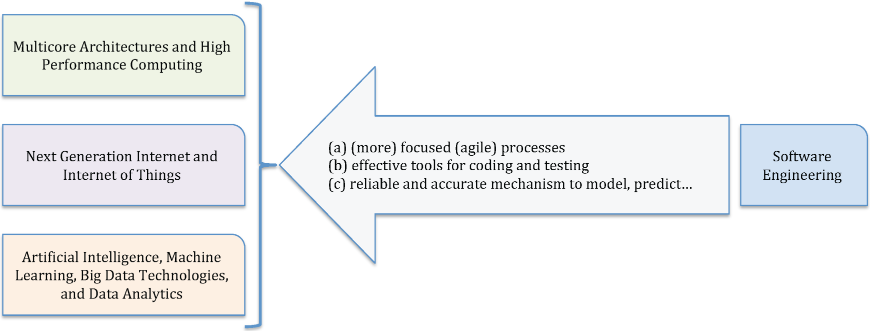


The distribution of ICSE/FSE papers in 2014, 2015, and 2016 in the "software verification and validation" category.

Essence of the results

- The industry needs better tools to predict software development and to understand the most recent proposals in terms of processes.
- The current effort in the area of Machine Learning and multicore development could provide a significant support to this.

Three areas of influence



Multicore Architectures and High Performance Computing

Next Generation Internet and Internet of Things

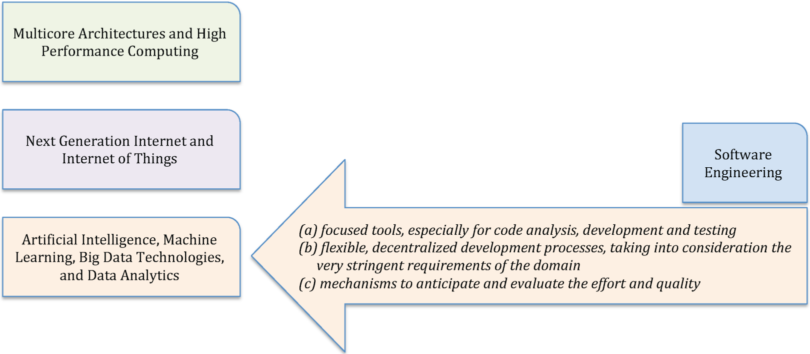
Artificial Intelligence, Machine Learning, Big Data Technologies, and Data Analytics

(a) (more) focused (agile) processes
(b) effective tools for coding and testing
(c) reliable and accurate mechanism to model, predict...

Software Engineering

The three main aspects in which Software Engineering can provide benefits to the major technological trends.

Focus on Data Science



Multicore Architectures and High Performance Computing

Next Generation Internet and Internet of Things

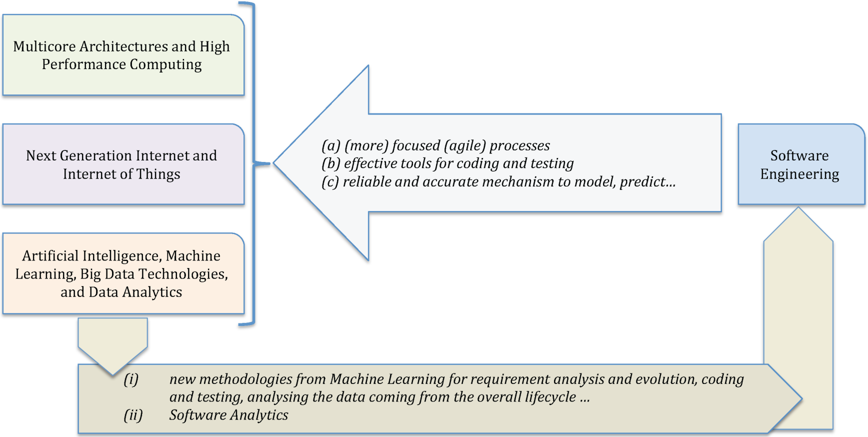
Artificial Intelligence, Machine Learning, Big Data Technologies, and Data Analytics

Software Engineering

(a) focused tools, especially for code analysis, development and testing
(b) flexible, decentralized development processes, taking into consideration the very stringent requirements of the domain
(c) mechanisms to anticipate and evaluate the effort and quality

Main aspects in which Software Engineering can provide benefits to Data Science.

Benefits back to SE



Multicore Architectures and High Performance Computing

Next Generation Internet and Internet of Things

Artificial Intelligence, Machine Learning, Big Data Technologies, and Data Analytics

(a) (more) focused (agile) processes
(b) effective tools for coding and testing
(c) reliable and accurate mechanism to model, predict...

Software Engineering

(i) new methodologies from Machine Learning for requirement analysis and evolution, coding and testing, analysing the data coming from the overall lifecycle ...
(ii) Software Analytics

Data Science “feedback” to Software Engineering.

Measuring progress in the work

Connecting measures in GQM

GQM approach

What Should We Measure?

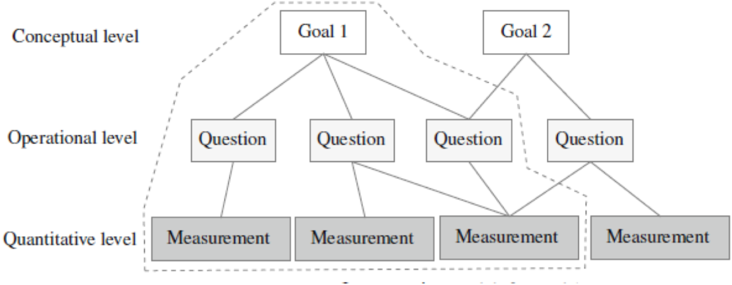
To answer this question we should answer the next questions:

- What we want to study? **Goal**
- How we want to study it? **Question**
- What should we measure? **Metrics**

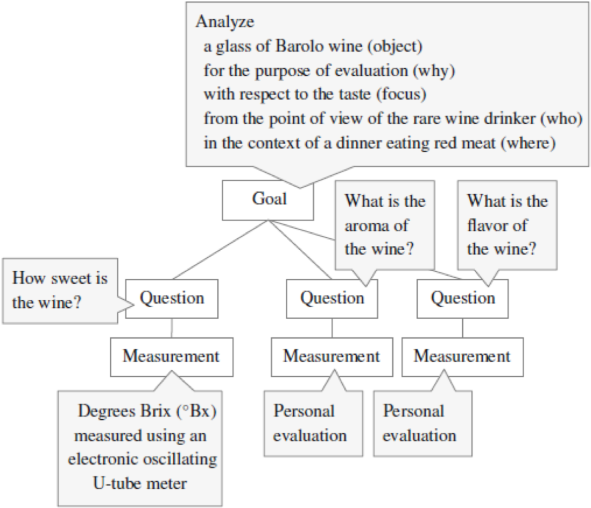
GQM approach

The GQM is is defined on three levels:

- Conceptual level (goal)
- Operational level (question)
- Quantitative level (metric)



Example

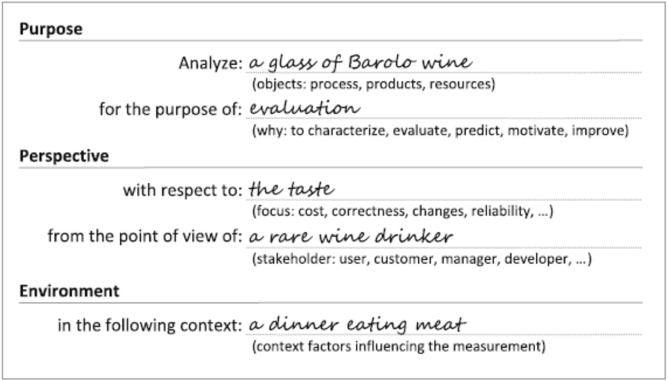


Analyze
a glass of Barolo wine (object)
for the purpose of evaluation (why)
with respect to the taste (focus)
from the point of view of the rare wine drinker (who)
in the context of a dinner eating red meat (where)

Goal
How sweet is the wine?
What is the aroma of the wine?
What is the flavor of the wine?
Question
Question
Question
Measurement
Measurement
Measurement
Degrees Brix (°Bx) measured using an electronic oscillating U-tube meter
Personal evaluation
Personal evaluation

Example Cont.

The definition of measurement goals is critical to the successful application of the GQM approach. To ease the definition of measurement goals, the GQM supplies goal templates



Purpose

Analyze: a glass of Barolo wine
(objects: process, products, resources)

for the purpose of: evaluation
(why: to characterize, evaluate, predict, motivate, improve)

Perspective

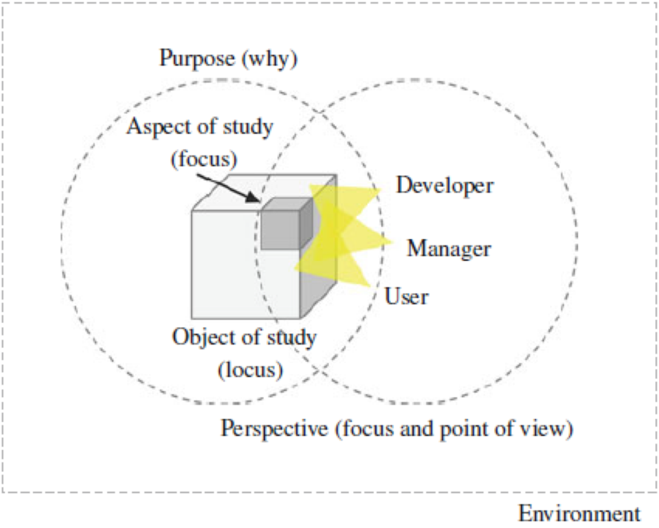
with respect to: the taste
(focus: cost, correctness, changes, reliability, ...)

from the point of view of: a rare wine drinker
(stakeholder: user, customer, manager, developer, ...)

Environment

in the following context: a dinner eating meat
(context factors influencing the measurement)

Example Cont.

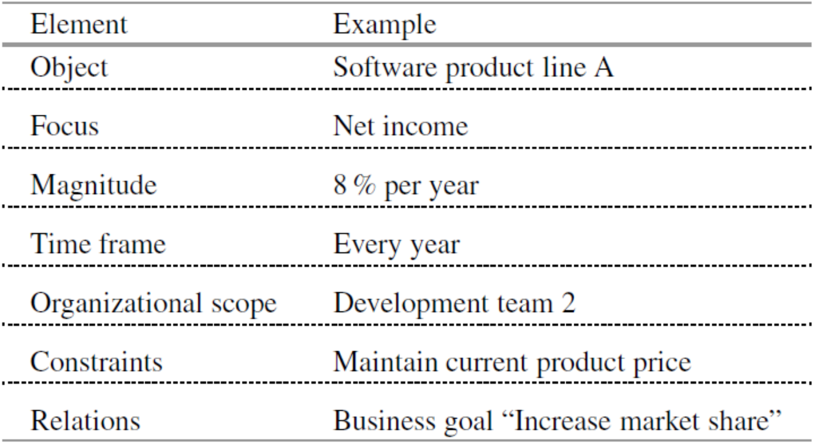


GQM+Strategies

- Object: the object of study;
- Focus: the aspect of study;
- Magnitude: the desired magnitude of improvement;
- Time frame: the time frame for achieving the goal;
- Organizational scope: the scope of responsibility for achieving the goal;
- Constraints: constraints or conflicting goals;
- Relationships: relationships to other goals.

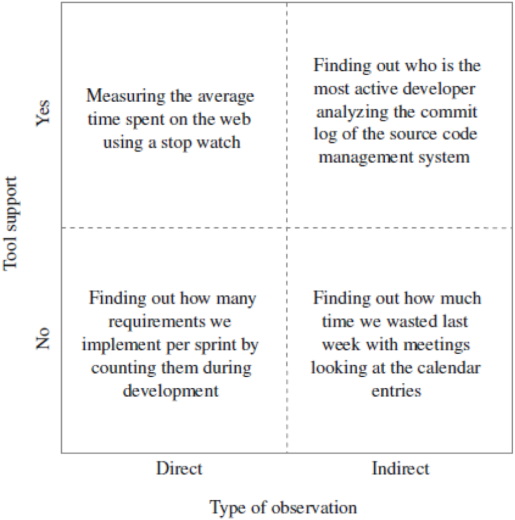
GQM+Strategies

business goal example



Element	Example
Object	Software product line A
Focus	Net income
Magnitude	8 % per year
Time frame	Every year
Organizational scope	Development team 2
Constraints	Maintain current product price
Relations	Business goal “Increase market share”

GQM approach



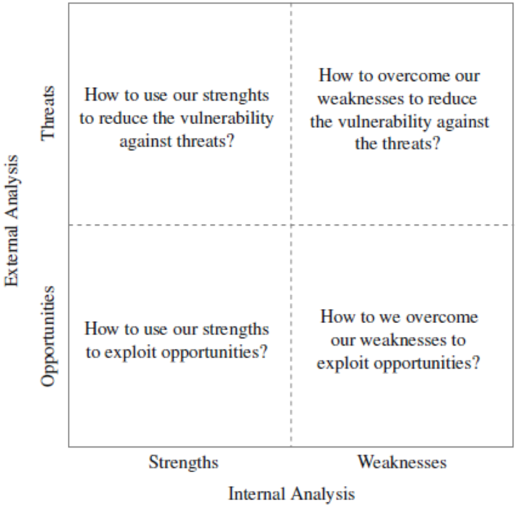
Tool support	Direct	Indirect
Yes	Measuring the average time spent on the web using a stop watch	Finding out who is the most active developer analyzing the commit log of the source code management system
No	Finding out how many requirements we implement per sprint by counting them during development	Finding out how much time we wasted last week with meetings looking at the calendar entries

Type of observation

SWOT analysis

- Strengths: organizational aspects that support the achievement of the objective;
- Weaknesses: organizational aspects that inhibit the achievement of the objective;
- Opportunities: external conditions that support the achievement of the objective;
- Threats: external conditions that inhibit the achievement of the objective.

SWOT analysis



External Analysis	Threats	How to use our strenghts to reduce the vulnerability against threats?	How to overcome our weaknesses to reduce the vulnerability against the threats?
	Opportunities	How to use our strengths to exploit opportunities?	How to we overcome our weaknesses to exploit opportunities?
		Strengths	Weaknesses
		Internal Analysis	