TYPES OF RESEARCH QUESTIONS

Feasibility:

- Does X exist and what is it?

Example: Is there any method that can tell whether to perform a software change?

- Is it possible to do X at all?

Example: Is it possible to build a system that can merge software change performed by two developers?

Characterization:

- What are the characteristics of X?

Example: Are Web services suitable for real-time communication?

- What exactly do we mean by X?

Example: What is the meaning of undesirable interactions in Web services?

- What are the varieties of X and how are they related?

Example: What are the flavors of the Scrum applied in the real-time software industry? Do they have common characteristics?

Method/mean

- How can we do X?

Example: How can we estimate the bug density that will result from a software change?

- What is a better way to do X?

Example: Known effort estimation methods are not accurate. How to improve them?

- How can we automate doing X?

Example: How can we automate version control?

Generalization

- Is X always true of Y?

Example: Are UML use cases always efficient for capturing customer requirements?

- Given X, what will Y be?

Example: Given an accurate measure of the components' reliability, can we estimate the whole system's reliability?

Discrimination

- How do I decide whether X or Y?

Example: How to decide whether to use Scrum or XP, given the application area?

TYPES OF RESEARCH RESULTS

Qualitative & descriptive models

- Report interesting observations

Example: Programming and learning skills of students increased since we adopted Scrum

- Generalize from real-life examples

Example: We surveyed 100 Web software managers. 97% of them assert that the MVC model increased their teams' productivity. We conclude that MVC increases productivity.

- Structure a problem area; ask good questions

Example: To better understand why most of the software teams do not use all the UML diagrams, the following question should be answered instead: how do those teams document their designs?

Techniques

- Invent new ways to do some tasks

Example: New protocol for Web services

- Implementation techniques

Example: New java design pattern

- Develop ways to select from alternatives

Example: Method to decide whether to perform a software change based on the component's reliability, bug density, and the performance requirements.

System

- Embody result in a system

Example: System that implements a new version control algorithm

Empirical models

- Develop empirical predictive models from observed data

Example: the analysis of data we gathered from 57 real-time software development organizations allowed us to elaborate the following maintenance effort estimation model:

Effort (man.month) = 0.0123 KLOC (1-bug-density)

Analytic models

- Develop structural models that permit formal analysis

Example: Logical theory of telephony services. Formal analysis: design defects translate in unsatisfiable formulas

TYPES OF RESEARCH VALIDATION

Persuasion

Example: We are designing a new Internet telephony protocol. We didn't finish. Right now, we implemented … and it works fine.

Implementation

Example: Here is an implementation of a version control system that uses the new algorithm

Evaluation

Example: The design pattern X is more flexible than Y

Analysis

- Formal model

Example: Erroneous service designs are proved to be unsatisfiable.

- Empirical model

Example: On a XEON dual-core machine, 4GB RAM, 4 MHz, the system capacity equals 24000. This result is much better than the previous algorithm that reached a capacity of 18000 on the same server machine.

Experience (case studies)

- Qualitative model

Example: Most of the surveyed programmers prefer Scrum because of the following 3 advantages: ...

- Decision criteria

Example: Overload control algorithm X outperforms algorithm Y in terms of goodput

- Empirical (statistical) model

Example: 93% of the surveyed developers assert they never used UML.