An Emerging Theory on the Interaction Between Requirements Engineering and Systems Architecting Based on a Suite of Exploratory Empirical Studies

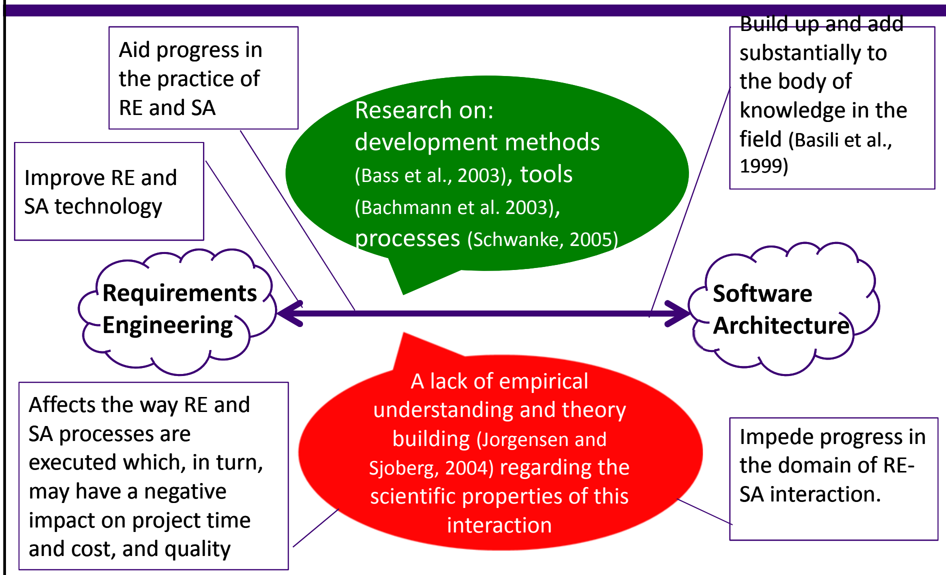
Unsing heroes:
Remo Ferrari and **Ibtehal Noorwali**

Western
UNIVERSITY · CANADA

Overview

9th July, 2014 (c) N.H. Madhavji, 2014 2

Motivation



Motivation

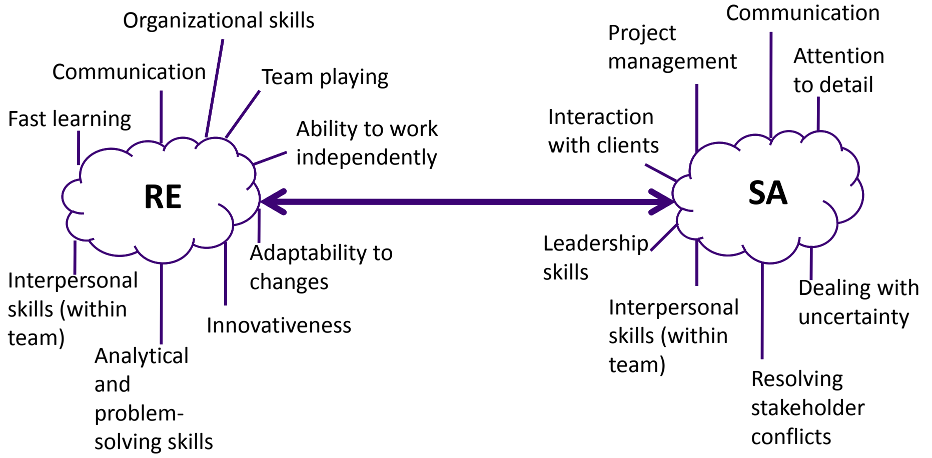
- Both technical and human aspects are considered critical for the success of software development (Bass and Berenbach, 2008).
- Human factors are even more important for RE and SA because these processes are at the front-end of the development cycle and are more aligned with real-world issues (Nuseibeh and Easterbrook, 2000).

Related Work

- Relationship Between RE and SA
  - Hints can be found in the literature on:
    - the need to consider the existing system in the RE process (Kotonya and Sommerville, 1998);
    - architecture's influence on requirements prioritisation and evolution (Nuseibeh and Easterbrook, 2000);
    - the iteration between RE and SA modelled as "Twin Peaks" (Nuseibeh, 2001);
    - NFRs in Architecting decisions making (Ameller, et al. 2013);
    - effects of design decisions on requirements elicitation and prioritisation (Durdik et al., 2013);
    - the effects of a customer/supplier relationship on stakeholders and architects (Poort et al., 2012)
  - Lack of theories in this area of SE is a clear sign of weakness in the field.

Related Work

- Non-Technical Factors in RE and SA (Ahmed, 2012), (Clements et al., 2007)



Overview of the Six Empirical Studies

Study ID	Study Title	Research Questions	Key Findings/Results
S1 (Ferrari and Madhavji, 2007)	Impact of RE Knowledge and Experience on Software Architecting	While architecting a software system, how do the architects with software requirements knowledge and experience compare against those without such knowledge and experience?	-The RKE groups developed a better architecture than the groups without RKE by an average of 10%. -The mental capability of the architects (regardless of RKE vs. non-RKE split) was the highest determiner of architecture quality (significant at p=0.003 or a 99.7% confidence interval). -Through regular feedback sessions with the architecting teams, the non-RKE group sought more feedback than the RKE group in quality-related categories such as tactics, quality scenarios, the satisfaction of quality, and pattern determination.

Overview of the Six Empirical Studies

Study ID	Study Title	Research Questions	Key Findings/Results
S2 (Ferrari and Madhavji, 2008b)	Requirements - oriented problems while Architecting	What kinds of requirements-oriented problems are being experienced while architecting a software system?	-Approx. a third of all problems faced by an architecting team were requirements-oriented problems. -There were several different types of requirements-oriented problems, of varying severity, which the architects faced while architecting: •Quality Satisfaction (22%); •Requirements understanding (18%); •Quality drivers determination (15%); •Abstraction (14%); • Modelling quality requirements (scenarios) (12%)

Overview of the Six Empirical Studies

Study ID	Study Title	Research Questions	Key Findings/Results
S3 (Ferrari et al., 2010)	Requirements Characteristics in the Presence/Absence of an Existing SA	-Which requirements characteristics were affected, and to what extent, by the presence or absence of the SA? -Which specific aspects of the SA affected the requirements?	-SA group analysts elicited technologically-oriented requirements; whereas, the non-SA group analysts elicited approx. 10% more user-needs oriented requirements. -For a given type of group (SA or non-SA), there is an inverse relationship within their set of elicited requirements between the characteristics "technological needs" and "user needs." -Specific architectural aspects were identified to have affected requirements characteristics: existing hardware, non-functional characteristics (same sub-system), non-functional characteristics (different sub-system), architectural patterns, and modifiability.

Overview of the Six Empirical Studies

Study ID	Study Title	Research Questions	Key Findings/Results
S4 (Miller et al., 2010)	Impact of Existing SA on Requirements Decisions	- How does an architecture affect requirements decision- making? - Which aspects of the architecture affect requirements decisions?	-Specific types of architectural effects on requirements decisions were identified: enabled (30%), constrained (25%), influenced (6%), and none (39%). -Approx. 60% of the requirements decisions were affected by the architecture. -Different aspects of the architecture were identified that had an impact on requirements decisions (e.g., NF characteristics, existing hardware, communication protocols).

Overview of the Six Empirical Studies

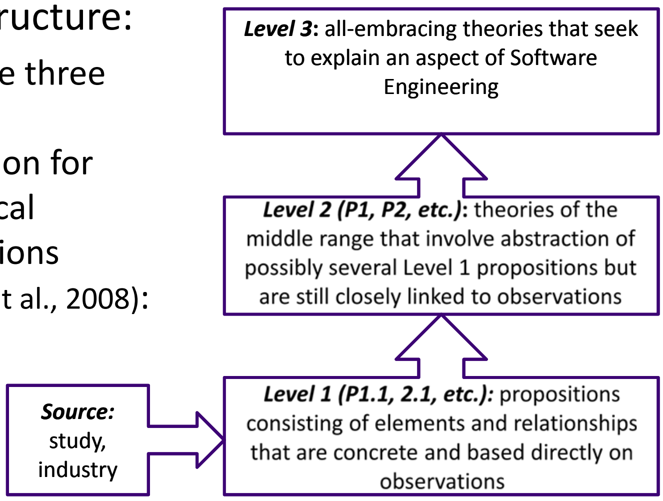
Study ID	Study Title	Research Questions	Key Findings/Results
S5 (Ferrari et al., 2010b)	Impact of Existing SA on Requirements Decisions in a Large-Scale, Prototypical Context	- What is the impact of an existing system's architecture on RE decision making? - What are the characteristics of the affected decisions? - What is the impact of the affected requirements decisions on the resultant system and downstream development activities?	-35% of the decisions were affected by previous architectural decisions: constrained (23%) and enabled (12%); 72% of constrained decisions were classified as *consequential* (i.e., decisions that emerged as a consequence of implementing other requirements decisions). -28% of the consequential decisions were from architectural oversights made previously. -72% of the consequential decisions were mostly self-contained decisions within a single module, and were not considered problematic. -Project staff mentioned that having RE and SA interaction information would be useful, but with staff turnaround, they neglected information capture because it incurred a project delay risk.

Overview of the Six Empirical Studies

Study ID	Study Title	Research Questions	Key Findings/Results
S6 (Ferrari and Madhavji, 2009)	Impact of Non-Technical Factors on SA	What is the impact of non- technical factors on Software Architecture?	-Numerous non-technically oriented problems were found among the teams: missing/late for meetings (32%); procrastination (22%); poor planning and group strategy (15%); individuals delivering inadequate work (14%); other, such as, lack of leadership, communication issues, mistrust between team members (16%). -Out of five (of 15) teams that had the least number of non-technical problems, four of these were the highest performing teams in terms of final architectural quality. Conversely, three of the five teams that had the most number of non-technical problems had the weakest architectures. -The identified problems led to extra time and rework during the architecting process.

Emerging Theory

- Theory Structure:
  - There are three levels of abstraction for theoretical propositions (Sjoberg et al., 2008):



Emerging Theory

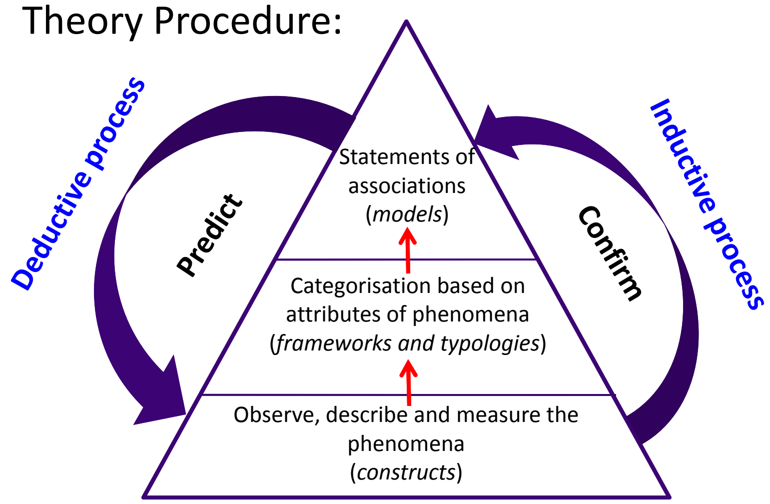
- Theory Procedure:
  - ***Abstraction*** from any given level to the next level considers the elements and relationships from the relevant lower level propositions so as to ensure their representation at the higher level proposition.

Emerging Theory

- Theory Procedure:
  - ***Construction process*** complements this three-level ***theory structure*** (Wacker, 1998):
    - conceptual constructs being investigated are identified;
    - observing the limits when and where events are expected to occur;
    - build relationships between observed constructs;
    - define the theoretical predictions and empirical support for the predictions.
  - This procedure is applicable to many types of research (e.g., empirical and analytical).

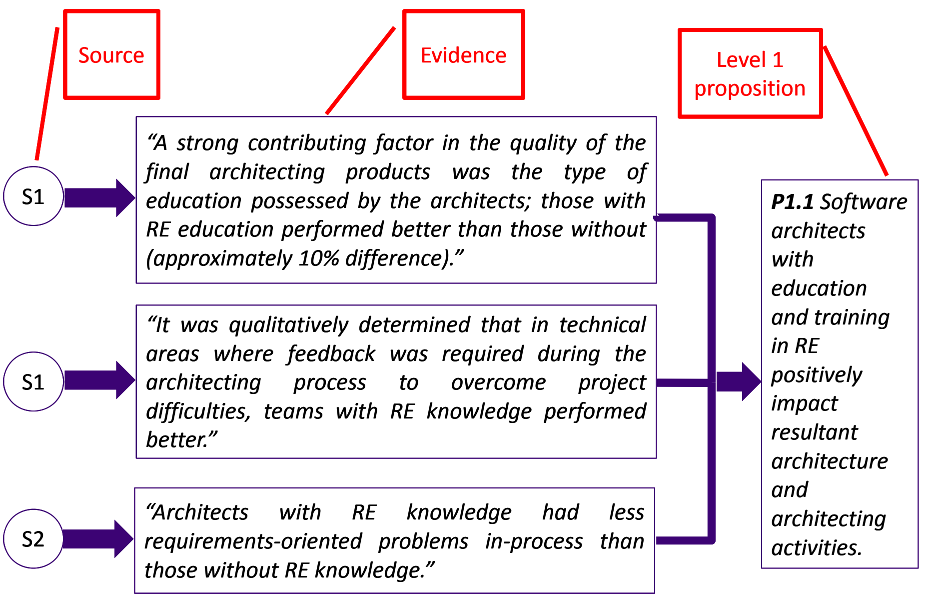
Emerging Theory

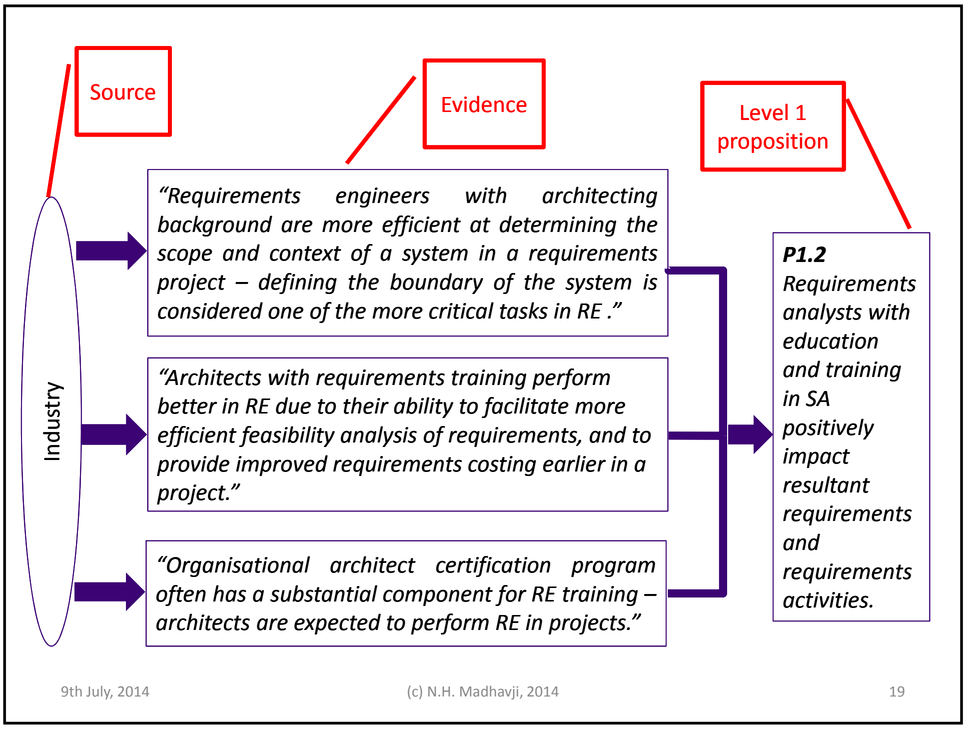
- Theory Procedure:

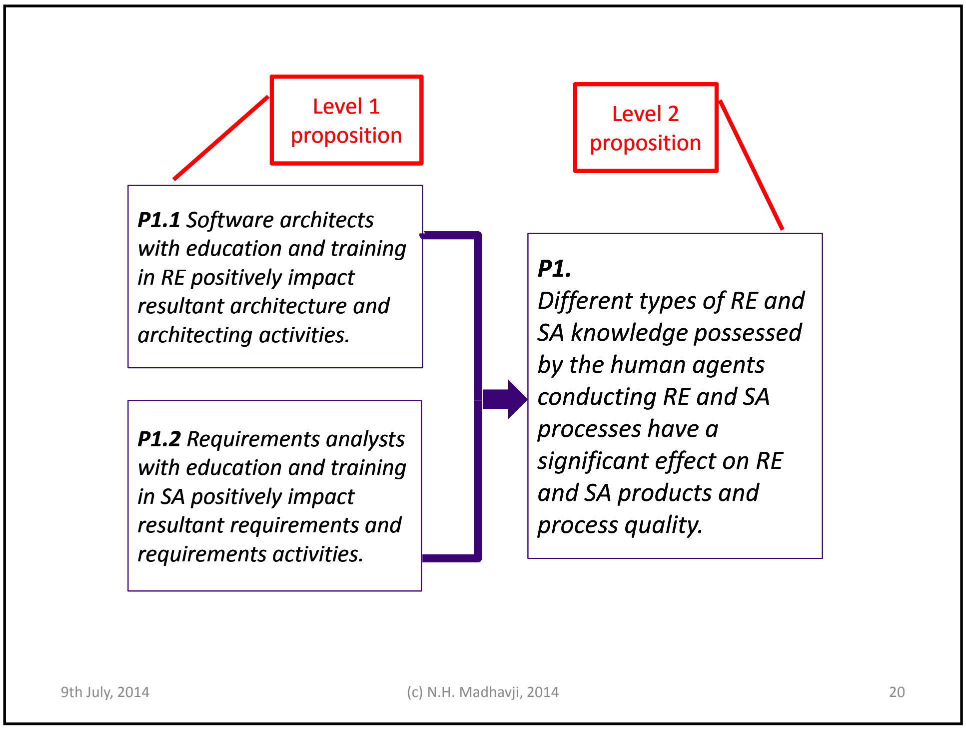


The Theory Building Process (Carlile and Christensen, 2005)

Emerging Theory: Human Factors



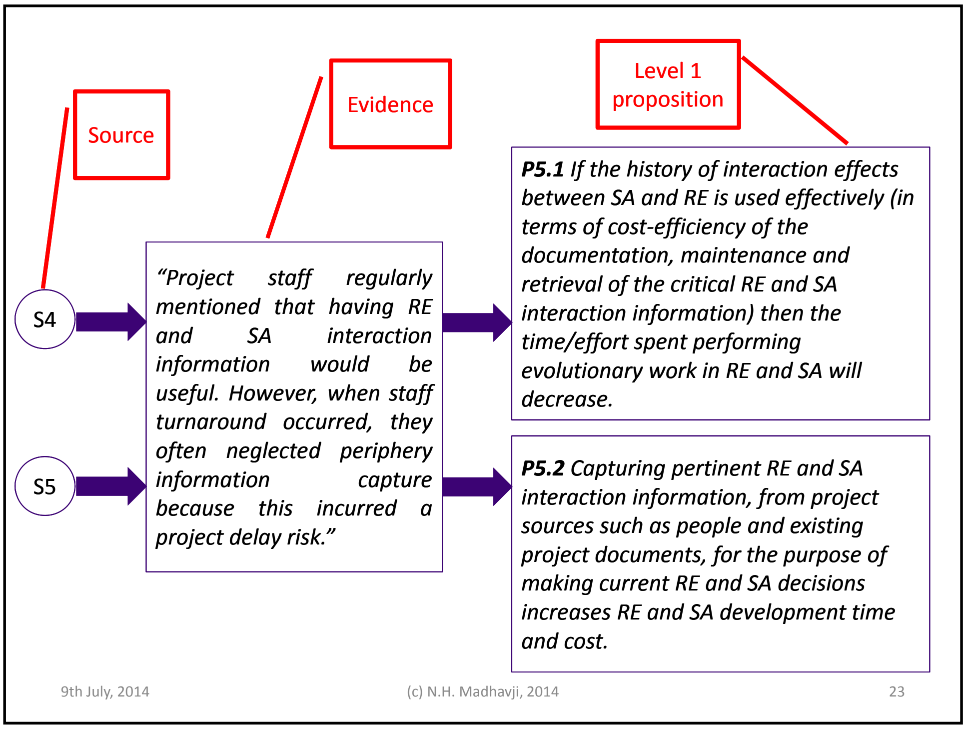

Source
Evidence
Level 1 proposition
Industry
"Requirements engineers with architecting background are more efficient at determining the scope and context of a system in a requirements project – defining the boundary of the system is considered one of the more critical tasks in RE ."
"Architects with requirements training perform better in RE due to their ability to facilitate more efficient feasibility analysis of requirements, and to provide improved requirements costing earlier in a project."
"Organisational architect certification program often has a substantial component for RE training – architects are expected to perform RE in projects."
P1.2 Requirements analysts with education and training in SA positively impact resultant requirements and requirements activities.
9th July, 2014
(c) N.H. Madhavji, 2014
19

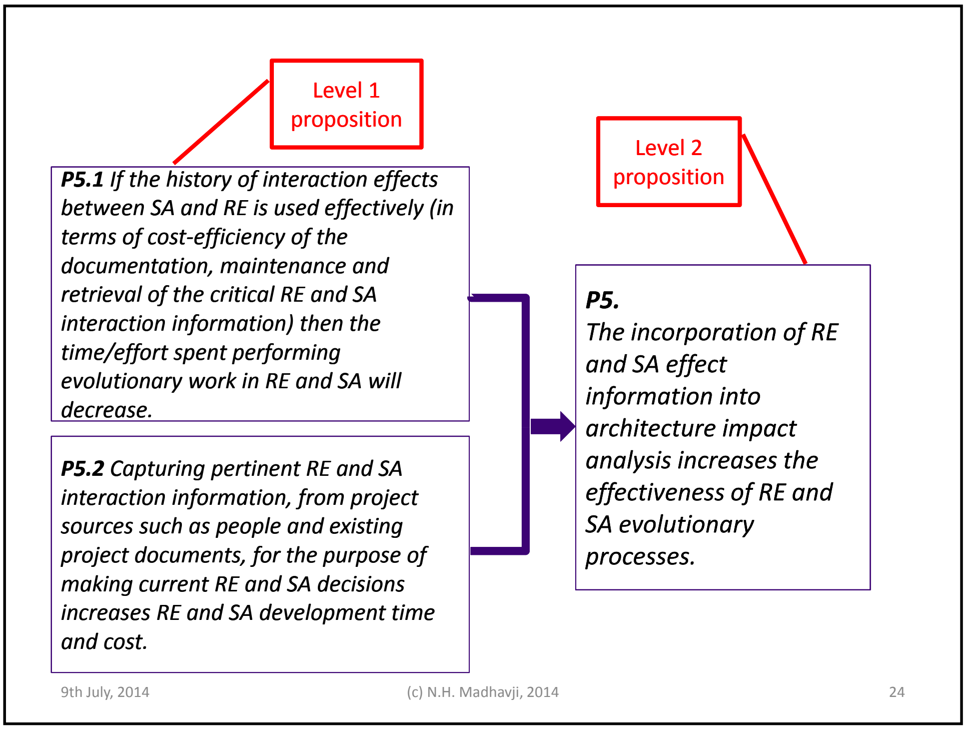

Level 1 proposition
Level 2 proposition
P1.1 Software architects with education and training in RE positively impact resultant architecture and architecting activities.
P1.2 Requirements analysts with education and training in SA positively impact resultant requirements and requirements activities.
P1. Different types of RE and SA knowledge possessed by the human agents conducting RE and SA processes have a significant effect on RE and SA products and process quality.
9th July, 2014
(c) N.H. Madhavji, 2014
20

Source	Level 1 Proposition	Level 2 Proposition
S3	**P2.1** A requirements elicitation team with motivation and expertise in system solution is more likely to elicit requirements that have technological bias.	**P2.** Varying types of skill-sets and personal interests (such as more technologically motivated vs. user-needs motivated) possessed by the human agents conducting RE and SA processes significantly alter resultant RE and SA product characteristics.
S3	**P2.2** A requirements elicitation team with motivation and expertise in a system's context (e.g., human- computer interaction and end-user satisfaction) is more likely to elicit requirements that are user-focused.	
S1, S6	**P3.1** Non-technical factors training and education reduces non-technically oriented problems in RE and SA.	**P3.** Human factors such as mental capability, education, and others (e.g., professionalism, communication, leadership, etc.) significantly override the impact of technological usage in an RE and SA project.
S1, S6	**P3.2** Mental capability, education and experience are the highest determinant factors for predicting RE and SA product quality.	
S1	**P3.3** The use of RE and SA technology does not significantly decrease variance between project outcomes in terms of RE and SA product and process quality.	

9th July, 2014 (c) N.H. Madhavji, 2014 21

Emerging Theory: Process Factors


Source
Evidence
Level 1 proposition
S4
S5
"Project staff regularly mentioned that having RE and SA interaction information would be useful. However, when staff turnaround occurred, they often neglected periphery information capture because this incurred a project delay risk."
P5.1 If the history of interaction effects between SA and RE is used effectively (in terms of cost-efficiency of the documentation, maintenance and retrieval of the critical RE and SA interaction information) then the time/effort spent performing evolutionary work in RE and SA will decrease.
P5.2 Capturing pertinent RE and SA interaction information, from project sources such as people and existing project documents, for the purpose of making current RE and SA decisions increases RE and SA development time and cost.
9th July, 2014
(c) N.H. Madhavji, 2014
23

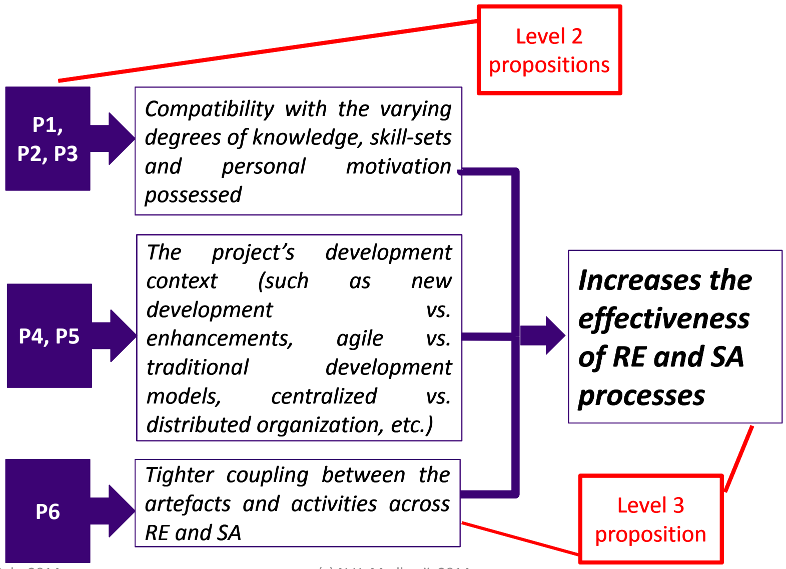

Level 1 proposition
Level 2 proposition
P5.1 If the history of interaction effects between SA and RE is used effectively (in terms of cost-efficiency of the documentation, maintenance and retrieval of the critical RE and SA interaction information) then the time/effort spent performing evolutionary work in RE and SA will decrease.
P5.2 Capturing pertinent RE and SA interaction information, from project sources such as people and existing project documents, for the purpose of making current RE and SA decisions increases RE and SA development time and cost.
P5. The incorporation of RE and SA effect information into architecture impact analysis increases the effectiveness of RE and SA evolutionary processes.
9th July, 2014
(c) N.H. Madhavji, 2014
24

Source	Level 1 Proposition	Level 2 Proposition
S4, S5	**P4.1** Non-functional (NF) characteristics of a non-local sub-system significantly affect (enable or constrain) requirements for the local sub-system being worked on.	**P4.** RE and SA artefacts and processes significantly vary when conducted in the presence or absence of an existing SA.
S4, S5	**P4.2** Constrained requirements decisions have a (strong) negative impact on construction and testing.	
S5	**P4.3** Constrained requirements decisions have a (moderate) negative impact on a multitude of project-specific system properties (such as performance, safety, reliability, etc.)	
S5	**P4.4** Older, "load-bearing" components of a system lead to more constrained effects on new requirements decisions than newer implemented components.	
S5	**P4.5** Significantly more *consequential* requirement decisions affect the SA than do core or emergent requirements decisions.	
S3	**P4.6** Requirements elicited from a RE process that involves analysis of a current architecture will be more technologically focused than a RE process that does not include such analysis.	

Source	Level 1 Proposition	Level 2 Proposition
S3	**P4.7** Requirements elicited from a RE process that does not analyze the current architecture will be more user-focused than a RE process that does not include such analysis.	**P4.** RE and SA artefacts and processes significantly vary when conducted in the presence or absence of an existing SA.
S3	**P4.8** Requirements elicited when the current architecture is analyzed are considered more important for system success (in terms of providing essential value for system stakeholders) than without such analysis.	
S3	**P4.9** Requirements elicited when the current architecture is analyzed are more architecturally relevant than requirements without such analysis.	
S3	**P4.10** The degree of requirements characteristics will vary between projects, but the impact from presence/absence of SA will be roughly the same.	
S4, S5	**P4.11** The existing architecture has a significant impact on new requirements decisions as a constraint or an enabler.	
S4, S5	**P4.12** Approximately 20-30% of requirements decisions are *constrained* by an existing SA.	

Source	Level 1 Proposition	Level 2 Proposition
S2, S4, S5	**P6.1** Coordination between requirements analysts and architects during handover processes reduces number of problems during RE and SA activities.	**P6.** RE and SA processes that are augmented with sub- activities that enforce a tighter integration between critical RE and SA links will lead to an increase in the effectiveness and efficiency of these processes.
S2, S4, S5	**P6.2** Requirements-oriented problems encountered during architecting are predominantly limited to *quality satisfaction, quality drivers determination, modeling quality requirements, abstraction, and requirements understanding.*	
S2, S4, S5	**P6.3** If the requirements engineers and software architects together model quality requirements, then the number of requirements-oriented problems during the architecting process will decrease.	
S2, S4, S5	**P6.4** If adequate background information about the requirements (such as, rationale, assumptions, priority, etc.) is given to, or shared with, the software architects then fewer requirements-oriented problems will be encountered by the architects.	
S2, S4, S5	**P6.5** If the architects provide "live" feedback to the RE agents on potential system-wide constraints and enablers, then the amount of requirements-rework will be reduced.	

Emerging Theory



Emerging Theory

- Theory Statement
  - Level 3 proposition (emergent theory) thus states that:
    - The **effectiveness** of **RE** and **SA** processes is **increased** if **technological support** ensures:
      1. **tighter coupling** between the artefacts and activities across RE and SA,
      2. **the project's development context** (such as new development vs. enhancements, agile vs. traditional development models, centralized vs. distributed organization, etc.), and,
      3. **compatibility** with the varying degrees of **knowledge, skill-sets and personal motivation** possessed.

Evaluation of Emerging Theory

- Boehm and Jain (Boehm and Jain, 2006) and Sjoberg et al. (Sjoberg et al., 2008) list similar criteria for evaluating the "goodness" of SE theories, both lists being adapted from other disciplines such as Business Management, Psychology, and Sociology.

Evaluation of Emerging Theory

1. **Empirical Support:** The degree to which a theory is supported by empirical studies that confirm its validity.
2. **Utility:** The degree to which a theory supports the relevant areas of the software industry.
3. **Generality:** The breadth of the scope of a theory and the degree to which the theory is independent of specific settings.
4. **Parsimony:** The degree to which a theory is economically constructed with a minimum of concepts and propositions**.**
5. **Testability:** Evaluates the degree to which a theory can be empirically refuted.
6. **Explanatory Power:** The degree to which a theory accounts for and predicts all known observations within its scope, is simple in that it has few *ad hoc* assumptions, and relates to that which is already well understood.

Evaluation of Emerging Theory

1. **Empirical support:** *low* to *moderate* → because more families of studies still need to be conducted in the domain of RE and SA interaction.
2. **Utility:** *high* → because all the propositions have a direct practical impact (on decision making in RE and SA projects).
3. **Generality:** *low* to *moderate* → because the empirical evidence on which the theory is derived is mostly from "lab" settings and does not extensively consider important RE and SA issues such as economic decisions, technology, team structure, and development lifecycle .
4. **Parsimony:** *high* → the expansive set of constructs is reduced to a smaller, more manageable number that improve clarity and understandability of the theory, effectively leading to its easier application to practice.
5. **Testability:** *high* → because each proposition of the emerging theory is expressed in a way that is directly testable
6. **Explanatory power:** *low* → because the theory is emerging and is based mostly on observations from exploratory studies.

Theory Statement

- The effectiveness of RE and SA processes is increased if technological support ensures:
  1. ***tighter coupling*** between the artefacts and activities across RE and SA,
  2. ***the project's development context*** (such as new development vs. enhancements, agile vs. traditional development models, centralized vs. distributed organization, etc.), and,
  3. ***compatibility*** with the varying degrees of ***knowledge, skill-sets and personal motivation*** possessed.

Implications

- Practice:
  - Employing architects with ***adequate background in requirements*** can reduce training costs and architectural defects.
  - Analysis and categorisation of ***architectural effects on RE decisions*** could help architects separate the more easily implementable requirements from the more difficult ones, leading to ***better***:
    - ***Project planning*** (e.g., time-to-implement, resource allocation, requirements prioritisation and scheduling),
    - ***Risk management*** (e.g., implementability), and ***product evolution*** (e.g., new feature planning).
  - Encouraging analysts to ***consider the existing SA*** when making RE-related decisions to ***avoid unnecessary rework*** and reduce rework costs.
  - ***Achieving business goals*** by predicting RE and SA project outcomes based on ***control/manipulation of key attributes***.

Implications

- Methods and tools in RE and SA :
  - Requirements management ***tools*** (such as DOORS and Requisite pro) and goal-oriented modeling tools (such as i* and KAOS) could be ***enhanced with a product- centric knowledge base***, which, in turn, could enrich SA tools (e.g., ArchE, Software Architect, etc.).

Implications

- Research:
  - Researchers can use the preliminary theory for hypothesis forming and testing, which can then be fed back into the theory.
  - Researchers can perform further grounded theory building on new research issues that were not explored in our studies.
  - Examples:
    - Controlled studies for establishing causality between non-technical factors and the success of a SA project.
    - Case studies to examine ***the effect on SA*** of issues such as ***team-members personality compatibility, team's heterogeneous skill sets, and, team politics and trust***.
  - The emerging theory may aid in assessing the ***maturity*** of the ***RE and SA interaction field*** (and its theory).

Future Work

- The proposed theory requires more empirical studies to:
  - test specific aspects of the theory,
  - expand the breadth of propositions that are currently described,
  - provide more detailed explanations as to why phenomena observed are occurring.
- This requires an effort by the RE and SA community to conduct studies in various contexts (i.e., "lab" or industrial practices) and to feed the resultant findings back into the evolution of the emerging theory.

Conclusions

- The closeness and importance of the RE-SA relationship has been clearly recognised in the literature.
- While advances in technologies have been made to aid transitioning from RE to SA, progress is slow on empirical work on the RE-SA interaction (human and process factors).

Conclusions

- There are no known empirically derived theories on RE-SA interaction, even though theories are foundational to the success of any scientific discipline (Basili et al., 1999).
- Based on the observations from six empirical studies, we propose a novel emerging theory as a small but important step towards filling the void.

Conclusions

- The theory is constructed "bottom-up" with traceability from lower-level observations to higher-level propositions of the theory.
- The emerging theory was evaluated based on "theory goodness" criteria and was found to satisfy criteria as follows:
  - utility, parsimony, testability (high);
  - empirical support and generality (low to moderate);
  - explanatory power (low).

References

F. Ahmed, “Software requirements engineer: an empirical study about non-technical skills,” Journal of Software, vol. 7(2), pp. 389-397, 2012.

F. Bachmann, L. Bass, M. Klein, “Moving from quality attribute requirements to architectural decisions,” Second International Workshop from Software Requirements to Architectures (STRAW ‘03). Portland, USA, pp. 122-129, 2003.

V. R. Basili, F. Shull, F. Lanubile, “Building knowledge through families of experiments,” IEEE Trans. On Software Engineering, vol. 25(4), pp. 456-473, 1999.

L. Bass, B. Berenbach, “Leadership and management in software architecture (LMSA'08) a report on an ICSE workshop,” ACM SIGSOFT Software Engineering Notes. vol. 33(4), pp. 27-29, 2008.

L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, 2nd ed. Addison-Wesley, 2003.

B. Boehm, A. Jain, “An initial theory of value-based software engineering,” In Value-Based Software Engineering, 1st Ed., S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, G. Grünbacher, Eds. Springer, Germany, 2006, pp.15-33.

P. Carlile, C. Christensen, “The Cycles of Theory Building in Management Research,” Harvard Business School Working Paper, No. 05-057, February 2005.

P. Clements, R. Kazman, M. Klein, “Working session: software architecture competence,” Proc. of the Working IEEE/IFIP Conf. on Software Architecture (WICSA). Mumbai, India, 2007.

Z. Durdik, A. Koziolek, R. Reussner, “How the understanding of the effects of design decisions informs requirements engineering,” Proceedings of the 2nd TwinPeaks Workshop. San Francisco, CA, pp. 14-18, 2013.

R. Ferrari, N.H. Madhavji, “Software architecting without requirements engineering knowledge and experience: What are the repercussions?” Journal of Systems and Software. vol. 81(9), 2008a. [Also appeared in Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'07), pp. 16, 2007]

R. Ferrari, N.H. Madhvaji, “Architecting-problems rooted in requirements”, Special Journal Issue of Information and Software Technology on Best Papers from REFSQ’05 and ’06, V ol. 50(1,2), pp. 53-66, 2008b.

R. Ferrari, N.H. Madhavji, “The impact of non-technical factors on software architecture,” 2nd workshop on Leadership and Management in Software Architecture. Vancouver, Canada, pp. 32-36, 2009.

R. Ferrari, J. Miller, N.H. Madhavji, “A controlled experiment to assess the impact of system architectures on new system requirements,” Requirements Engineering Journal, vol.15(2), RE'09 Special Issue. K.T. Ryan, Eds. pp. 215-233, 2010.

References

R. Ferrari, O. Sudmann, J. Geisler, C. Henke, W. Schaefer, N.H. Madhavji, “Requirements engineering decisions in the context of an existing architecture: a case study of a prototypical project,” In Proc. of the 18th IEEE Requirements Engineering Conf. Sydney, Australia, pp. 79-88, 2010b.

M. Jorgensen, D. Sjoberg, “Generalization and theory-building in software engineering research,” 8th International Conf. on Empirical Assessment in Software Engineering (EASE 2004), pp. 29 – 35, January 2004.

G. Kotonya, I. Sommerville, Requirements engineering - processes and techniques. Wiley, 1998.

J. Miller, R. Ferrari, N.H. Madhavji, “An exploratory study of architectural effects on requirements decisions,” Journal of Systems and Software, vol. 83(12), pp. 2441-2455, 2010.

B. Nuseibeh, “Weaving together requirements and architectures,” IEEE Comp. vol. 34(3), pp.115-117, March 2001.

B. Nuseibeh, S. Easterbrook, “Requirements engineering: a roadmap”, Proceedings of the Conf. on the Future of Software Engineering, ACM Press, pp. 35-46, 2000.

E. R. Poort, A. Key, P.H.N. de With, H. van Vliet, “Issues dealing with non-functional requirements across the contractual divide,” Proc. of the 10th Working IEEE/IFIP Conf. on Software Architecture and 6th European Conf. on Software Architecture (WICSA/ECSA). Helsinki, Finland, pp.315-319, 2012.

R. Schwanke, “GEAR: a good enough architectural requirements process,” 5th Working IEEE/IFIP Conf. on Software Architecture (WICSA 05). Pittsburgh, USA, pp.57-66, 2005.

C. Shekaran, “Panel overview: the role of software architecture in requirements engineering,” Proceedings of 1st International Conference on Requirements Engineering. pp. 239, April 1994.

D. Sjøberg, D. Dybå, B. C. Anda, J. Hannay. “Building theories in software engineering,” in Guide to Advanced Empirical Software Engineering. F. Shull, J. Singer, D. Sjøberg, Eds. Springer, 2008, pp. 312–336.

J. G. Wacker, “A definition of theory: research guidelines for different theory-building research methods in operations management,” Journal of Operations Management, vol.16, pp.361-385, 1998.

Thank you!
شـــكــــــرا

Q & A