Chris Cargile 10/13/2013
CSCI 602 - - Foundations of SWE (Dr. M. Verdicchio) Paper 2

Prompt 1: Interpretations of a Requirements Engineering Roadmap

Nuseibeh and Easterbrook, two prominent figures in the realm of published works in the field of Requirements Engineering (RE) since before 2000, provided a general framework upon which the notions of requirements engineering could be conceptualized, formalized, and further developed, according to multiple influential factors. A structure for the paper and how to conceptualize the details it presents is provided along with mention of the structure used in other RE literary works for doing so, at the beginning of the paper. The authors commence by giving a definition of the overall field and approaches applied to satisfy it, from which they can then develop the ideas presented in greater detail. In providing a generalized definition of RE then explaining how the chosen definition is apt for framing ones understanding of RE, the authors valuate the definition's applicability before transitioning into deeper assessments of the foundational constructs shaping RE. In giving a context by which to understand RE, the authors allude to an explanation that could debatably be argued is befitting the broader discipline of software engineering: a system being designed to serve a constituency's needs can only be as effectively designed as per the constituency's ability to convey said needs in that Nuseibeh, et al. acknowledge an unquestionable dependency on effective communication. The importance of communication is presented where RE is explained to be reliant upon an approach that is multidisciplinary in nature for the elicitation and forecasting of users' needs under evolvable conditions and such that RE draws upon scientific approaches for addressing real-world issues faced by *humanity* as well as engineering-based approaches useful for evolving systems in a *deterministic* fashion. The shape of a multi-disciplinary approach needed, as presented by [1] has been argued to apply to SWE as a whole by [3].

Zave's definition concerning RE is dismissed quickly by highlighting that precision is too rigid a term for a specification heavily interleaved with the beliefs and understandings expressed by human users, and the authors move quickly into explaining techniques based on the social sciences as a means for eliciting and modeling requirements, once again returning to a multidisciplinary context in which to view RE. That RE is best practiced with a means that is continual and integrative in nature and that RE may also easily be likened to a potentially front-loaded endeavor would provide a difficult structure for the paper to adhere to, as these ideas might seem contradictory in some ways. Perhaps for this reason, the authors provide some additional explanation and illustration of why and how RE can be useful. Depending upon the knowns and unknowns(/risk) and the knowables and unknowables of a project, RE's benefits, practicality, and process can differ, and the authors allude to this as a notion that parallels the description of RE given by Finkelstein. In addition to risks involved and nature of a development effort, creating functional specifications involves documenting key aspects of the design that includes the boundaries, stakeholders, and goals for the developed system upon the project's completion.

When reading through [1] the reader might feel that several underlying points are recurrent. It may seem in many ways the authors are attempting to afford clarity towards matters which may have seemed to present a problem in the history of analyzing RE, particularly as to the undertaking of explaining the role requirements engineers (RE's) play for helping the business and how they contribute to the overall organizational process. The authors seem intent on emphasizing, possibly as an attempt to explain the role of RE's to inquisitive business leaders, the humanistic purposes RE's play and on arguing that their underlying function may be one involving identification and facilitation of common, *high-level goals* and that in myriad ways RE's serve to help prevent shortcomings resulting from communication breakdowns. The impression that the authors assign this as a role to RE's might be derived based upon the statement from [1] that "users [often] find it difficult to articulate their requirements," and it could be extruded that keeping an RE involved could aid in circumventing breakdowns in a continuing, iterative-type approach whilst the RE helps to identify patterns or *scenarios* which fit the needs expressed by the customer, throughout the entirety of a developmental effort.

The authors are able to adequately convey that the RE is a quintessential participant during extraction, documenting, and development phases during the building-out of new requirements-based features. Under 4.2, the elicitation techniques described reiterate a need for precision in a human-oriented context so as to obtain consensus and develop models of the components from which a designed system is comprised. The subsection asserts recent work in the area of RE elicitation techniques was directed at finding ways that approaches to building models of cognition could complement the ideals of the softer side of compensating the unique, individual trains of thought - or social dynamics - as well as environmental factors (contextualization), in a situation.

The authors continue by enlightening the reader that depending on the application domain, guidance on the appropriateness of an elicitation technique can be provided in the form of *methods*. [1] quickly returns, however, by revisiting the subject of modeling, with an introduction to the purposes of modeling and by advising that an explanation of the means for analyzing them is forthcoming with brief overviews of various modeling techniques to follow, as well. From beyond 4.3, the paper could be considered to teeter on the underlying themes represented earlier in their work, involving a human-oriented- versus a prescriptive, universally acceptable, and finite- approach to requirements and focusing on providing details of the modeling and other aspects of RE. It could be observed the paper concludes with the observation that goals and consensus take a higher priority than the modeling of system state and information flow while asserting, yet another time, that there remain proponents who insist on the prioritization of contextualization as a factor influencing system design.

The authors do not make mention of any losses or gains towards contextual human inputs versus one-size-fits-all models of the Human responses to environmental conditions, except they mention that there exist a range of methods – from soft to formal – in terms of methods available to capture contextual cues affecting a model. And they mention the work of Loucopoulos through which enterprise modeling is likened to teleology – based on the idea that nature [requirements models] tends towards definite ends - so the authors may be suggesting some believe formal methods of modeling are appropriate whilst others believe, still, that softer modeling approaches are more fitting.

Between the end of the paper in section 10 and beyond the point at which it begins to revisit earlier visited underlying themes (consensus; softness of approach vs. formalness of approach, as per section 4.3), the authors transition their description of the nature of RE by describing that a system’s requirements goals must remain communicable, consensus-representing, and traceable to ensure their long-term viability and/or survivability. Overall, a history of RE is provided in their work and the paper leaves the reader more knowledgeable of the systematic and practical aspects influencing the roles of an RE and of the ways in which RE helps achieve the end-goals of stakeholders-at-large.

References:

[1] Bashar Nuseibeh and Steve Easterbrook. 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (ICSE '00). ACM, New York, NY, USA, 35-46. http://doi.acm.org/10.1145/336512.336523

[2] T. H. Tse and L. Pong. 1991. An examination of requirements specification languages. *The Computer Journal*. 34, 2 (April 1991), 143-152. http://dx.doi.org/10.1093/comjnl/34.2.143

[3] Vinton G. Cerf. 2012. Where is the science in computer science?. Commun. ACM 55, 10 (October 2012), 5-5. DOI=10.1145/2347736.2347737 http://doi.acm.org/10.1145/2347736.2347737

PROMPT 2: R.E.: A Roadmap Including Elicitation Techniques Compared to Case Study-Recommended Practices

Nuseibeh, et al. [1] presents an analysis of the current, past, and future states of the discipline of Requirements Engineering (RE) in a pattern of: inferences being made from observations based in academic research, a pattern familiarly referred to as a research roadmap. [1] provides a sound foundation upon which to analyze the theoretical and substantive tradeoffs from among different approaches to SWE/RE being applied industry-wide at present and of the past and, in conclusion, present several challenges that lie ahead for applications of it in the future.

It could be observed that Nuseibeh, et al. [1] provide a contextual definition to construct boundaries according to which differences in the benefits and tradeoffs associated with various RE approaches can be assessed, and alternatively that Napier[2] takes a more presumptive, yet more finite approach by foregoing any contextualized explanation as to the meaning of RE in defining it. Rather, Napier proceeds to detail several methods available for assessing and improving upon currently proposed RE processes. While the different purposes exist for the differently styled papers, Maibaum provides an observation that is constructive in the effort for gaining insight into why SWE/RE is of importance. Maibuam [3, pg.3] projects that SWE/RE is an engineering discipline, logically, in the sense that: "Engineering refers to the practice of organizing the design and construction of any artifice which transforms the physical world around us to meet some recognized need." This definition affords a lens through which to view RE's meaning such that more-developed analyses of RE can be applied to the Roadmap by [1] and the RE Process Assessment by [2].

Aside from the general purposes [1] and [2] are written, there is not only a similarity between them on the basis of a belief that RE can make achieving real-world objectives possible with an empirically backed, process-driven approach - which transforms the world around us into terms a computer can interact with, ie: via modeling. [1] and [2] are in concurrence that adhering to even a loosely-defined process for analyzing the necessary considerations in a development effort can aid one's understanding of how best to apply RE lessons learned.

If [1] provides a general background (intro, foundations, context and groundwork) upfront, in their literary work, and proceeds by detailing how to elicit, model, analyze, communicate, agree, evolve and integrate requirements, [2] begins similarly, by providing background, then proceeds by detailing the lessons learned from a case-study based methodology. Both [1] and [2] evolve into conclusively mentioned areas for future exploratory effort to grow the field; while [1] provides the reader insights as to the course RE as a whole may tend towards, and [2] indicates ways by which stakeholder perceptions and process prescriptions can be combined during assessments to effectively inform RE process improvement.

In turn, [2] addresses how to implement a balanced approach to RE assessment by considering inductive and prescriptive modeling approaches to RE assessment; while [1] provides a broader overview of assessment approaches by considering the various elicitation techniques (4.2). There does not exist an absolute, vis-à-vis connection between the two papers as per the description of RE assessment by each, but it is worthwhile to consider their similarities and differences and to make inferences based on the guidance they provide to better understand RE and as to when which particular practices might be best utilized.

Napier[2] addresses three challenges discussed in RE assessment process literature, making mention of steps taken in a case-study and declares the combinatory approach to assessment implemented in their study was effective as a result of: careful management of interactions between stakeholders and developers between cycles; its inherent, advantageous strength via offering a dual-edged and systematic approach to RE assessment; and partly due to circumstances specific to the client examined during the industrial case study. There is not a declaratively proposed condition by which to assign proper balance between prescriptive- vs. inductive approaches to assessment for any particular circumstance or company. Maibaum[3] contends that the majority of software based systems could be built using normal design principles, but does not propose a means by which to elicit or to capture formally stated requirements, nor how to gauge stakeholder consensus upon them, as is a product of requirements process assessment. In

Sections 2 and 3 from [2], however, pros and cons associated with inductive (problem-based) and prescriptive (model-based) approaches to assessment are explored. [2] provides details of the activities involved with a dual-edged approach to assessment, starting with pre-assessment then post-assessment activities, followed by overall lessons-learned. Situated between the detailing of tradeoffs between inductive and prescriptive approaches and the application of the actual techniques employed in their case study, [2] provides a description of how the candidate assessed came to be chosen and as to how their action-research effort was made possible, defining concrete means by which to qualify assessment approaches.

An attempt to further identify a comparison between the two papers cannot be easily made by reviewing the details of [2]'s action-research effort and of how [1]'s assessing of the different approaches to elicitation unfolded, but we make an effort to do so, below, then return to discussing areas for further research suggested by both.

To conduct the RE assessment, [2] describes an approach that initiates with the investigators essentially deciding to address challenges discussed in RE assessment-process literature by splitting the focus on inductive vs. prescriptive assessment approaches in a 75/25 split, respectively - primarily based upon the candidate's previous experience applying approaches. [2]'s interest in identifying a balance between specificity, comparability, and accuracy challenges in RE process improvement addresses the chasm that exists between the potential to apply engineering design principles to resolve software engineering shortcomings and the current status such is that few companies have actually applied formal assessment processes by which software can be systematized.

To provide insight on the structuring of the papers, the implementation plan chosen by the PST and applied by the researchers, as mentioned in [2] is described. It involved four pre-assessment inquiry cycles, followed by recommendations made based on the previous inquiry cycles and the implementation of the recommendations across two cycles. The post-assessment phases, consisting of describing the lessons-learned during the previous phase of inquiries and through another iteration of the inquiry-cycle processes, after the implementation of the recommendations proposed by the PST, segues the case study findings' direction into overall reflections, which reveals that the 75/25 split between inductive and prescriptive approaches to assessment was useful in their case.

[1] takes a broader-encompassing stance on assessing the pros and cons of the various approaches to assessment than [2]; while, to illustrate the approaches used in each literary work in succession, [1] not only describes, succinctly, five different approaches to RE in general terms, it further infers there are important points to consider when assessing whether a process assessment approach should be more model-driven and less cognitive-/contextual- (or user-) driven versus the contrary - a more inductive-driven approach.

[2] suggests that software process improvement and RE assessment could stand to benefit from a systematic approach that is more inclusive of stakeholder interests as well as one which provides a uniform metric against which to compare the business in the context of other industry businesses' competitive standing. Yet, [2] does identify some areas of weakness related to the applicability and reproducibility of their findings in a more generalized sense. Maibaum asserts a systematization and better designs are perhaps achievable using [formal] design principles, if more companies implemented them [3, pg.8]. However, a potentially important observation is that many deliberators have pondered whether software engineering as a discipline can maintain its categorization as an engineering discipline based on the transformations of user beliefs to systems predicated on theorems of certain types of logic.

[1] discusses various approaches to elicitation but does not resolutely stand behind a particular one, explicitly, but in the conclusion section notes that, after the 90's the ideas that modeling assessment approaches should be considerate of stakeholders' and participants' perceptions -and that models are never absolutely accurate, by their nature- emerged. A similarity between the indicated futures for RE assessment approaches will likely be uncovered by the astute assessor of current RE futures predicted.

References:

1) Bashar Nuseibeh and Steve Easterbrook. 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (ICSE '00). ACM, New York, NY, USA, 35-46. http://doi.acm.org/10.1145/336512.336523

2) Nannette P. Napier, Lars Mathiassen, Roy D. Johnson, "Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study," IEEE Transactions on Software Engineering, vol. 35, no. 5, pp. 593-606, September/October, 2009

3) Tom Maibaum. 2000. Mathematical foundations of software engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (ICSE '00). ACM, New York, NY, USA, 161-172. http://doi.acm.org/10.1145/336512.336548

Prompt 3: A summary of Cao and Ramesh

[1] discusses Agile Requirements Engineering (RE) and the results obtained from a two-phase, qualitative research study involving 16 organizations, which used the grounded-theory method to develop insights about agile RE without a prior hypothesis. The two-phase study involved collecting and analyzing data from the organizations according to which category that were matched – either as being on that adheres to XP, Scrum, or both or as following an RE approach that resembles one or both of them.

The authors address several of the often touted benefits of agile RE alongside the potential or observable challenges realizing such benefits might present in the perspective of organizations employing them in their business, in the presentation of their study's findings. Seven practices, in total, which are characteristic of the approaches employed are presented and the benefits and challenges of them are discussed in order of: face-to-face communication over written specifications; iterative requirements engineering; requirement prioritization; managing change in requirements through constant planning; prototyping; using test-driven development (TDD); and using review meetings and acceptance tests.

[1] asserts that agile is not definitively superior approach for managing the challenges associated with RE, as compared with traditional or more upfront RE approaches; yet, it does not demand the same upfront or ongoing risks. Rather, [1] contrasts the risks associated with the two approaches by drawing attention to the ways in which the benefits of the agile approach exceed those afforded by a traditional approach, as is explained from here:

The authors explain that face-to-face communication primarily affects better transfer of ideas from the customer to developers, a lesser burden of upfront documentation, and further ability for the customer to be more adaptable to changes. On the other hand, adaptability may be one of the few '–ilities' that an agile approach guarantees to address, and [1] admits if a communication breakdown does occur due to personal interaction or the lack of documentation maintained, the nature and extent of resultant problems can be diverse. With an iterative, agile approach to RE, customers may be more likely to get what they want – more business value-added software as from a process with less associated confusion – but this adaptive approach places additional, other pressures on the customer. Constant communication is vital; it should be understood that poorly-defined or nonexistent modeling upfront can be inherently risky; and sometimes large fluctuations in the estimated cost to deliver an implementation of previously unknown features can occur.

[1] dispels that test-driven development (TDD) is a regularly used process in many real-life organizations and that, instead, many organizations favor an approach that is more dynamic and less formalized. One section in which [1] denotes a possible reason for a lessened reliance on formal modeling is in the discussion of requirements prioritization; here, customers typically aid the development effort by stating those features which provide them the most benefit or business value. For this reason, efforts such as formal verification or detailed consistency checking rarely occur.

[1] leaves the reader with the impression that an agile approach is advantageous, as compared to an iterative one, in environments where requirements are ambiguous or constantly changing and where the costs and possible benefits associated with the agile practices in one's particular project environment can be understood ahead of time to enable gauging its contributed value. While [1] contends that certain practices claimed by others to be constructive, such as TDD and prototyping, may be so in theory but in practice are generally not the norm. It is reasserted that constant, effective communication is a key tenet to agile, and it is explicitly stated that developers may be able to alleviate some of the unnecessary responsibility for generating or maintaining documentation on features which may change before they provide value, while shifting some risks associating with requirements ambiguities onto the shoulders of the customer to aid in to how developmental efforts are perceived by the customer. Obtaining continued, clear communication while not setting an expectation for constant prototyping can be part of a recipe for success.

References:

1) Lan Cao and Balasubramaniam Ramesh. 2008. Agile Requirements Engineering Practices: An Empirical Study. *IEEE Softw.* 25, 1 (January 2008), 60-67. http://dx.doi.org/10.1109/MS.2008.1