Requirements Change is a Natural Law - Learn How to Live With It

Rick Dove
Paradigm Shift International and The Agile Security Forum
Box 289, Questa, NM 87556
dove@parshift.com, 505-586-1536

Panel discussion position, loosely stated:

- Requirements are perceptions about the nature of a problem.
- Like any attempt at modeling something in the real world, requirements form a discrete set of abstractions that attempt to encompass what is important.
- Requirements are constructed by people, presumably doing the best they can to perceive the true nature of the problem, and the best they can to judge when that perception sufficiently encompasses what is important.
- Complexity is the rule rather than the exception with problems of interest, and the consequences of complexity defy human perception.
- All perceptions are biased, some are more insightful than others, none are precise, most continue to evolve [1]. Ergo...requirements continue to evolve.
- Project unpredictability is principally rooted in requirements evolution.
- The commercial sector does not have better requirements-development practices than government-funded aerospace/defense, but perhaps they have learned to live with requirements change more effectively...

a. They may refit requirements to actual accomplishments more readily: Holding a contractor or an internal department to a set of requirements for contractual purposes at odds with new understandings or simple realities is not in the best interest of the people who make such decisions. Requirements are (relatively) easily respun for political and expediency reasons - neither the market nor management waits, and something is (often) better than nothing.
b. They may be more diligent in a less forgiving environment: market share and ROI is dependent on inflexible market windows. Also, there is evidence in Europe that lower and unforgiving budgets increase the recognition of good-requirements value.
c. They may be more agile in their approach: Learning how to dance with market uncertainty has become a commercial-sector necessity. Requirements development is more likely to value open options and risk management mitigation as an integral part of requirements development.

Real Requirements-Practice Examples - Objectives and Methods

Case 1 – Objective: Develop leadership transparency interface for customer outsourced-manufacturing WIP. Method: a) Collaborative discussions internally about breakthrough possibilities, b) customer engineer with superior customer knowledge and ownership takes responsibility to build customer interview questions, c) professional interviewers hired to facilitate customer interviews.

Case 2 – Objective: Changeable, mixed-vendor ERP system organized around an agile infrastructure. Method: a) Developed general requirements for system dynamics, b) held preliminary RFI discussions with vendor candidates, c) issued 11 page RFQ focused on dynamic requirements (functional requirements were undifferentiating among vendors, so ignored).

Case 3 – Objective: Predictable ERP integration process. Method: a) 3-phase implementation, b) requirements frozen for each phase, c) no customization of applications, d) vendor sole responsibility for functional results.

Case 4 – Objective: Cyber-security strategy consistent with: a) unprecedented web-accessible enterprise-wide data transparency, b) being a high probability political and economic target, c) operating in South-East-Asia ethical environment. Method: Employ top quality security advisors to develop requirement priorities.

Case 5 – Objective: Reduce fraud and mistakes in public-company financial reporting. Method: a) Sarbanes-Oxley non-compliance penalty legislation, b) clear emphasis on transparency as an objective, c) SEC directive to auditors to interpret and satisfy the intent of SoX anew for each audit rather than employ a static check list (enlightened realization: explicit regulatory requirements are backward looking and cannot anticipate the future).

Real Requirements-Practice Examples - Results

Case 1 – WIP transparency system: Ho/hum, yes/no, witness-leading Q&A did not probe for new possibilities, but instead confirmed preconceived conservative and protective thoughts of customer engineer. ... Useless results. Why?

Case 2 – Mixed-vendor ERP infrastructure: Recycled standard product descriptions did not address any of the stated requirements. ... Useless results. Why?

Case 3 – Frozen-requirements integration process: Lip-service cycles through multiple meetings increasingly used the right words without any change to integrator's standard process. Vendor chastised at every meeting to go back and try again - to no avail. Total disbelief in frozen-requirements approach. ... Useless results. Why?

Case 4 – Cyber Security: Standard risk assessment matrix for requirements development wants to reduce everything to quantifiable financial impact and event probabilities. Question: what is the cost in revenue if a customer's IP is stolen? A: Infinite...potentially death of company. Rebuttal: No, you have to give me a number so the math can sort priorities. ... Huh?

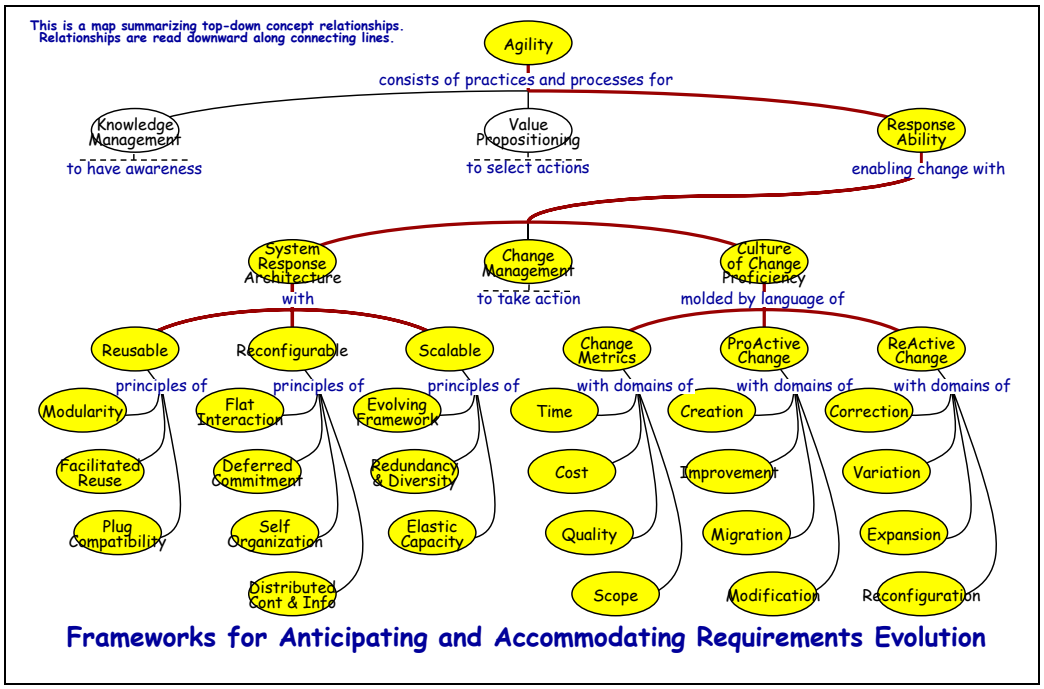
Case 5 – SoX Compliance: Consternation and confusion in the auditing community because judgment and initiative is required, and liability is not eliminated by check marks on a procedural list. ... No surprise here!

Framework-Driven Requirements Development

Cases 1 through 4 described above all had happy endings - but not until a framework-driven approach was employed that focused on requirements which recognized situation dynamics. These frameworks came from research into agile systems conducted in the nineties [2] and were purposefully applied for the first time to cases 1-4 above in the early 2000s.

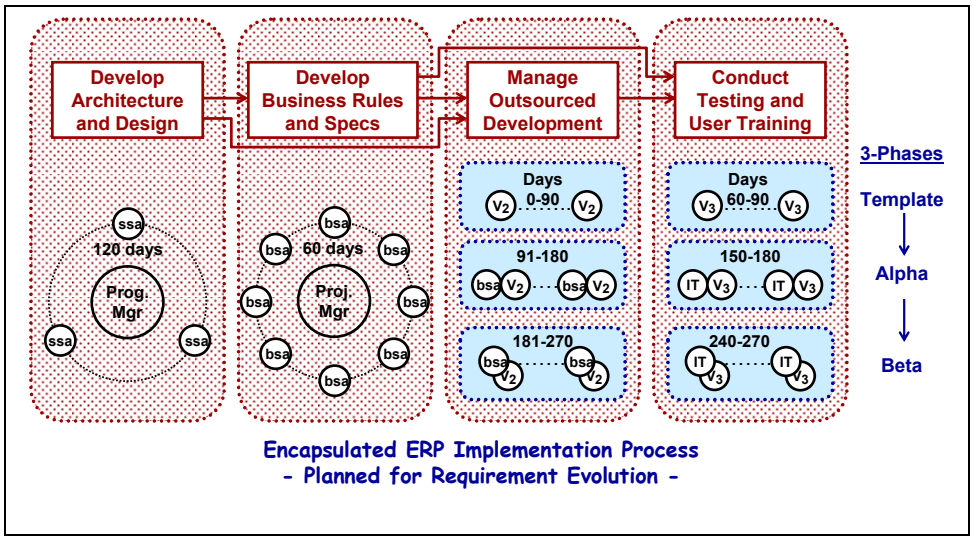
The framework forced consideration of requirements evolution in two general categories: proactive and reactive response to new knowledge. Each of these general categories had four sub-categories for consideration. Frameworks served to stimulate thought about potential requirements change. Another framework was employed to identify the critical response dynamics for each of the potential requirements change areas: response time, response cost, response quality, and response scope. The collection of proactive and reactive change issues and their critical performance metrics were then used to develop requirements for acceptable solution strategies - using a third framework of ten design principles organized around reusable modules reconfigurable in a scalable architecture.

Details of this framework driven approach were discussed for Cases 2 and 3 above at CSER 2005 [3]. A graphic representation of the Case 3 process approach is shown below. The environment of the Case 3 ERP implementation was a massive and highly unstable semi-conductor foundry startup in Malaysia. ERP is strongly driven by business process requirements - but this company had immature processes and no clear business process understandings or desires. It was clear that process rules would evolve for some time - on top of the fact that everything else about the environment was highly subject to change as attention moved from one crash startup project to another.



Frameworks for Anticipating and Accommodating Requirements Evolution

Situation Dynamics that Affected Requirements



Encapsulated ERP Implementation Process
- Planned for Requirement Evolution -

- Unstable company ($1.5 Billion massive start-up scramble)
- Unstable ERP (Oracle 11i: new, buggy, undocumented)
- Undefinable business processes (inexperienced company staff/mgmnt)
- Under experienced IT staff (Malaysian resource inadequacy)
- Bus vendor team (changed from Australian to USA)
- ERP vendor team (changed from USA to Malaysian)
- Planner Choice (changed from Oracle to Adexa)
- Planner system (added later)
- Time and Accounting system (added later)
- HRM system (added later)
- ETL design (evolved to template approach)
- CIM integration (major data integrity problems)
- Operational Transparency System (major data integrity problems to accommodate)

General Strategy

- Business System Analyst (BSA) Group: Assigned to IT-assist dept managers (cross dept responsibilities), Business Process IT application configuration/evolution, IT tool selection/acquisition
- Strategic System Analyst (SSA) Group: Evolution of infrastructure framework, Enforcing infrastructure usage rules
- User Collaboration: Mandatory response-dynamics requirements analysis
- COTS Applications Only: No customization of purchased software
- IT Internal Responsibilities – not to be outsourced: Infrastructure architecture design and evolution, Management of integration projects, Configuration of applications
- Vendor is responsible for total solution: HW and SW
- Requirements will not change during implementation
- No customization allowed
- Three Phase Implementation Sequence:
  P1: Out-of-box best practice from vendor – supporting the company
  Vendors configure the applications
  P2: BSA-developed business process rules
  Vendors + BSAs configure the applications
  P3: Refined business processes
  BSAs configure the applications
- No violation of infrastructure rules (repeatedly invoked)
- Don't say it can't be done, tell what is needed to do it (repeatedly invoked)

Results

- The company had functioning out-of-box (phase 1) ERP supporting the business within 90 days of implementation start, a custom business-process phase 2 implementation 90 days later, and a refined phase 3 implementation 90 days after that. Typical comparable implementations were taking 24-36 months, according to the Oracle implementers.
- The entire project was implemented on-time and below budget. Initial ERP applications were predominately Oracle 11i, as they had the only acceptable web-capability at the time. Licensing was budgeted at $5 million and implementation at another $5 million. Licensing was on budget and implementation came in close to $4 million. Comparable typical numbers were $15 to $25 million, according to the Oracle implementers. We were one of the first to implement Oracle 11i, meaning we wrestled with the usual new-software instability and bug discovery problems.
- A PeopleSoft HRM application collection was added in a 2-phase sequence: out-of-box and final. Three months were scheduled for each phase. The total for both came in at five months, against a comparable typical expectation of 12-18 months - according to the PeopleSoft implementers.

References

1. Dove, R., *Value Propositioning: Perception and Misperception in Decision Making*, Iceni Books, 2005.
2. Dove, R., *Response Ability: The Language, Structure, and Culture of the Agile Enterprise*, Wiley, New York, 2001.
3. Dove, R., "Fundamental Principles for Agile Systems Engineering," *Proceedings CSER 2005*, Stevens Institute of Technology, Hoboken, NJ, 2005.