Individualizing Requirements Information

David Gelperin
ClearSpecs Enterprises
david@clearspecs.com

1. Introduction

Requirements Engineering (RE) is about communicating needs, wants, and preferences during system development or change. RE should have its own requirements – specifically the types and forms of information to be communicated.

Information choices should be individualized to the many contexts (for example, stakeholder population, depth of understanding, and system criticality) associated with a project. Different types of systems (Jackson 2000, Kovitz 1998, Gelperin 2005) need different types of information.

Template tailoring is the commonly suggested approach to information selection, but is only useful when there is little variance in the customer base and the type of system. A more powerful approach is individualization using components. This paper describes a process for individualizing requirements information.

Different approaches to system development promote different types and forms of requirements information. **Traditional development** promotes comprehensive system specifications that is, lots of written information**. Agile methods** promote user stories, acceptance tests, and intense customer involvement to communicate requirements information that is, little written information. Both approaches provide packaged answers to questions about type and form. We will describe a third approach, supplier-driven development, that promotes a third package of answers.

Most descriptions of the requirements process (for example, Nuseibeh 2000, Wiegers 2000) omit the critical step of information selection. After identifying stakeholders and before selecting elicitation techniques, **one should select the types and forms of requirements information to be communicated**. The selected information should be "just enough" (that is, necessary and sufficient) to satisfy the essential information needs of all stakeholders while complying with budget and schedule constraints and providing information mandated by laws and standards. If this is not possible, conflicts should be identified and addressed early.

This paper describes a set of questions and activities for selecting a cost-effective set of types and forms. We begin by describing supplier-driven requirements and then provide a simple model that integrates it with traditional and agile approaches to help you begin the selection process.

2. Supplier-driven requirements

What do financial planning, computer assembly, and ecommerce website development have in common? In each, a reliable supplier knows a great deal about the details of a solution, has an inventory of components to solve a piece of the problem and has many questions to ask the customer. Accurate answers enable the supplier to assemble an individualized system that satisfies the customer's needs. Requirements (answers to supplier questions) can be viewed as parameter values for a produce-system function.

If an individualized solution is not necessary (tax preparation, spam blocking, and requirements management), the situation is also supplier-driven, but solved by off-the-shelf solutions.

Even when assembled or off-the-shelf solutions are not available, it is wise to seek out suppliers who have "done it before". Using a knowledgeable supplier results in significant cost and predictability advantages. Note that a free market economy will breed knowledgeable suppliers for systems having a profitable base of potential customers.

Consider the implications of a supplier-driven approach for requirements development, specification, and management as well as project estimation and testing. The supplier plays the role of analyst and is relied on to "know what to ask". Supplier questions and customer answers make up a requirements specification and may be documented in a contract or letter of understanding. There would be no requirements management in a traditional sense. Project estimates would be provided by the supplier and expected to be accurate. While a knowledgeable supplier reduces risk, some qualities (for example, maximum load) and system responses to misuse would still need to be thoroughly tested – perhaps with supplier-provided tests.

We now provide a model to start the information selection process.

3. Starting individualization

The following model uses three levels of stakeholder understanding to suggest starting points for individualization. While the values at the intersections of understanding (that is, in the boxes) are packaged answers to questions about information types and forms, they provide a fitting bias for the selection process.

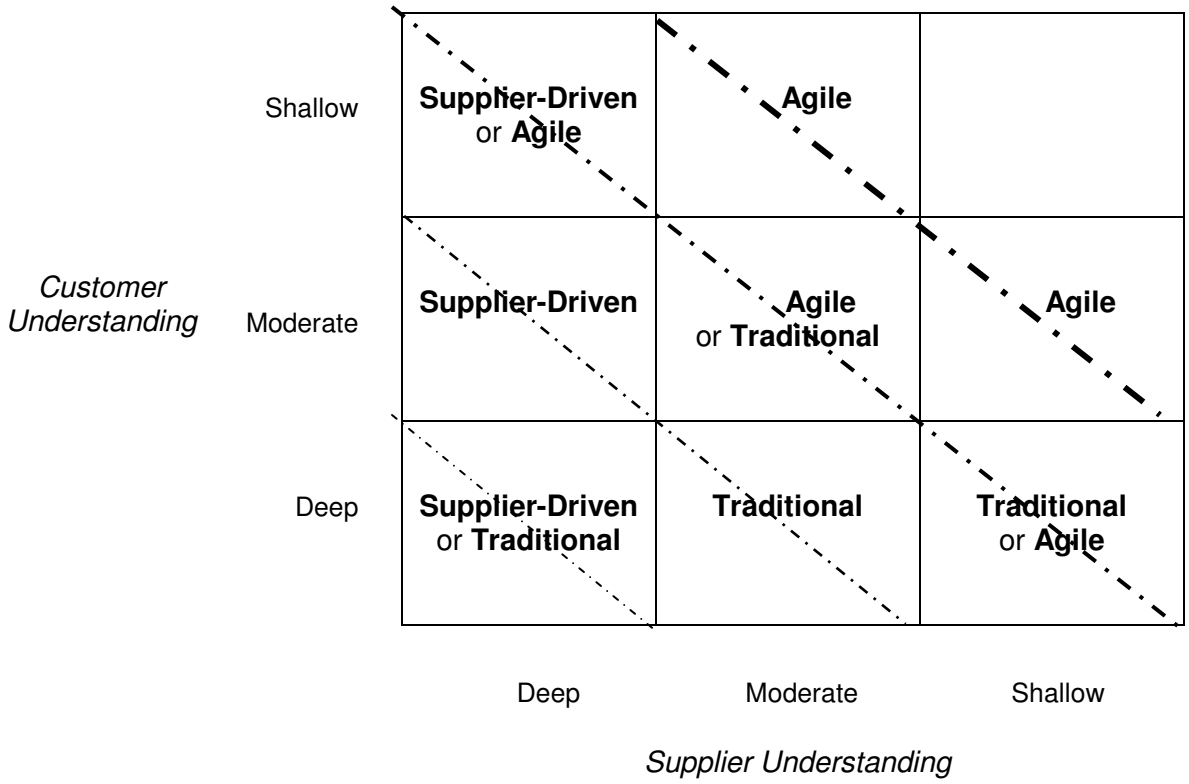

Figure 1: Starting points as a function of stakeholder understanding

Multiple choices in a box suggest that a blended strategy might be effective. Different levels of understanding about different parts a system suggest the use of different types and forms for the different parts.

The model contains bands of increasing risk moving from the southwest to the northeast. The riskiest box is intentionally blank.

In a perfect world, populated by truth tellers and truth hearers, people admit ignorance and fear thus making risk easier to manage with a clear separation between research and development.

We now consider the entire individualization process.

4. The individualization process

The following two tables provide lists of information types that could be useful on some project. The lists are comprehensive, but not complete, that is you may not find your favorite type or it may not have a familiar name. The lists are meant to help you create your own comprehensive menu from which your individual projects will make their selections. Downloads of these lists and descriptions of some of their entries (for example, quality specs) are among the technical papers at www.clearspecs.com.

Table 1: List of information types (Part 1)

A. Overview	A.2. Requirement Descriptions	A.2.2. Environmental Requirements
A.1. Background	A.2.1. System Requirements	A.2.2.1. Platforms
A.1.1. Introduction	A.2.1.1. Data Structures	A.2.2.2. Internationalization
A.1.2. Enterprise needs	A.2.1.2. Function Groups	A.2.2.3. Integration
A.1.3. Assumptions	A.2.1.3. Qualities	A.2.2.4. Interfaces
A.1.4. Alternative solutions	A.2.1.3.1. Safety	A.2.3. Operational Requirements
A.1.5. System summary	A.2.1.3.2. Security	A.2.3.1. Installation
A.1.6. Environmental impacts	A.2.1.3.3. Reliability	A.2.3.2. Operation
A.1.7. System justification	A.2.1.3.4. Usability	A.2.3.3. Access
	A.2.1.3.5. Scalability	A.2.3.4. Usage
		A.2.3.5. Documentation
		A.2.3.6. Training

Table 2: List of information types (Part 2)

A.2.4. Developmental Requirements	C. Behavior Models	F. Term Specs
A.2.4.1. Verification & Validation	C.1. Decision Tables	F.1. Entity Specs
A.2.4.2. Design Constraints	C.2. State Tables	F.2. Action Contracts
A.2.4.3. Implementation Constraints	D. Facts	F.3. Quality Specs
A.2.4.4. Project Constraints	D.1. Constant Conditions	F.4. Common Descriptions
B. Usage Models	D.2. Condition Dependencies	F.5. Acronyms
B.1. Context Diagrams	E. Derivations	
B.2. User Stories	E.1. Derived Values	
B.3. Use Cases	E.2. Derived Conditions	
B.4. Scenarios	E.3. Derived Actions	
B.5. Acceptance Test Specs		

You need to decide (1) which types of information satisfy essential information needs and (2) which forms (verbal; informal notes; structured text; formal expressions; diagrams; tables) provide optimal support for stakeholder decision making.

To select types and forms, you need answers to the following questions:

- Who are the project stakeholders – upstream (that is, customers, users, operators, testers) and downstream (that is, project managers, designers, developers, testers)?
- How consistent are the goals and visions of the upstream stakeholders?
- What type of system will be developed or changed?
- To what extent do the upstream stakeholders understand their needs and the characteristics of automated systems?

- To what extent do the downstream stakeholders have experience with and understanding of the type of system required?
- How do downstream stakeholders use requirements information?
- What is stakeholder experience with and tolerance for detailed specifications?
- How critical is system failure or delay?
- How extensive are safety and security requirements likely to be?
- Is model analysis necessary (for example, use case analysis)?
- What is the likely stability of the requirements information?
- How likely is the use of a standard architecture, off-the-shelf components, or a rapid development technology?
- What requirements information is mandated?
- What requirements information is affordably available?
- What requirements development and management tools are available?
- What are the likely information acquisition roadblocks and constraints?

While answers to some questions may not be available at the start of your project, the available answers should be factored into your choices of type and form. The answers should also be revalidated and supplemented as more is learned during the project. New or changed answers should cause a reassessment of current choices.

If requirements management is appropriate (agile or traditional), choose the requirements attributes to specify (see Table 3).

Tables 3: Inventory of requirement attributes

Identifier & Version	Supporting assumptions	Sources
Name	Rationale	Owner
Type & Subtype	Priority	Author
Groups	Security level	Status
Prerequisite reqts.	Verification strategy	Issues
Corequisite reqts.	Implementation effort	Comments
Derivative reqts.	Implementation priority	Change dates
Interacting reqts.	Implementation risk	System allocation

In summary, the process of individualizing requirements information includes:

- Survey stakeholders to answer the decision factor questions
- Identify stakeholders to lead the information selection process
- Use stakeholder understanding to identify a starting point
- With a starting bias, review information types and require a strong need for inclusion
- Choose forms for each selected type and identify responsible stakeholders and priority (compulsory; high; medium) as well
- Choose requirements attributes, if requirements management is appropriate
- Publish selections for stakeholder review
- Change selections as appropriate

5. Previous Work

Boehm and Turner (Boehm 2004), among others, contrast traditional and agile development. These discussions directly or indirectly include differences in requirements information. No discussion focusses on requirements information when suggesting criteria for choosing between approaches.

Kovitz (Kovitz 1998) describes differences in requirements information using Jackson's Problem Frames (Jackson 2000) as a reference system. System type is the only variable considered in this discussion.

Andrea (Andrea 2005) describes requirements individualization for projects that are not perfect for pure Agile methods.

Barry (Barry 2004) proposes an alternative information form.

6. References

Andrea, Jennitta "If the Shoe Doesn't Fit: Agile Requirements for Stepsister Projects" **Better Software** Vol. 7 No. 7 Sept. 2005 pp. 34-38

Barry, Daniel M. et. al. "User's Manual as a Requirements Specification: Case Studies" in **Requirements Engineering** Vol. 9 2004 pp. 67-82

Boehm, Barry and Turner, Richard **Balancing Agility and Discipline: A Guide for the Perplexed** Addison-Wesley 2004

Gelperin, David "Using Requirements Specification Patterns" 2005 Available among the technical papers at www.clearspecs.com

Jackson, Michael **Problem Frames: Analyzing and Structuring Software Development Problems** Addison-Wesley Professional 2000

Kovitz, Benjamin **Practical Software Requirements** Manning Publications 1998

Nuseibeh, B. A. and Easterbrook, S. M. "Requirements Engineering: A RoadMap" in A. C. W. Finkelstein (ed) **The Future of Software Engineering** (Companion volume to the proceedings of ICSE'00). IEEE Computer Society Press 2000

Wiegers, Karl "When Telepathy Won't Do: Requirements Engineering Key Practices" **Cutter IT Journal**, May 2000