Concretizing the Vision of a Future Integrated System – Experiences from Industry

Rikard Land, Ivica Crnkovic, Stig Larsson

Mälardalen University, Department of Computer Science and Electronics
PO Box 883, SE-721 23 Västerås, Sweden

{rikard.land, ivica.crnkovic, stig.larsson}@mdh.se, http://www.idt.mdh.se/{~rld, ~icc }

Abstract. *When an organization faces new types of collaboration, for example after a company merger, there is a need to integrate the existing software. Important challenges are how to create a realistic vision of a future integrated system, how to make the vision concrete enough to be able to work towards the vision, and of course to carry out the actual integration process. This paper focuses on how to concretize the vision.*

We have carried out a multiple case study, consisting of 9 cases. This paper presents the observations made in the form of recurring patterns that can be used as recommendations for other organizations facing the same challenge.

Keywords. Software Evolution, Software Integration.

1. Introduction

From time to time within an organization, two or more in-house developed software systems address similar needs, and there is an overlap in functionality. This typically happens when the organization changes through new types of collaborations and mergers. The organization would ideally want to take the best out of the existing systems and integrate them with as little effort as possible. This could for example mean reusing components of the systems in a new system, integrate them more loosely, discontinuing one system and extending the other, or even discontinuing both and start development of a new generation. See Figure 1.

We assume there is a vision of a future, integrated system. It is typically not difficult to outline a vision at a very high ("PowerPoint") level, but this vision must be concretized – otherwise it will only have the (non-)shape of a mirage. This is not to say that the vision must be completely defined before starting actual integration (the cases indicate rather the opposite).

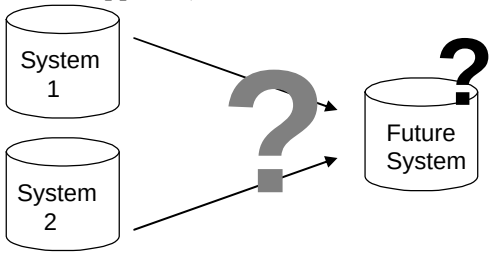

Figure 1: Two challenges – the future system and the path there.

Based on a previous case study [2,4] we have assumed three main influencing factors: processes, architecture, and requirements/ features. We intend to analyze each of these factors thoroughly in other publications, while the present paper outlines the relationship between them.

The relationship between the vision and these three influencing factors can be simplified described as follows: input to the vision are 1) the architectures of the existing systems, 2) the features of the existing systems, and 3) the available resources and desired timeline for the integration. Of these, the first two are fixed (they describe the actual existing systems) while the third is re-negotiable. The vision must then be concretized into the corresponding descriptions of the future system: architecture, features (or requirements), and a project plan. See Figure 2.

The challenge is then to follow the integration process plan and iteratively adjust the vision and concretize it into (slightly modified) architectural descriptions, feature descriptions of the future system, and an integration process plan. Reality soon becomes too complex for this simple model to be useful for anything else than an initial description of

the concepts. The actual analysis in the present paper will therefore be in the form of "patterns", i.e. activities that will aid in this iterative process of making the vision more concrete.

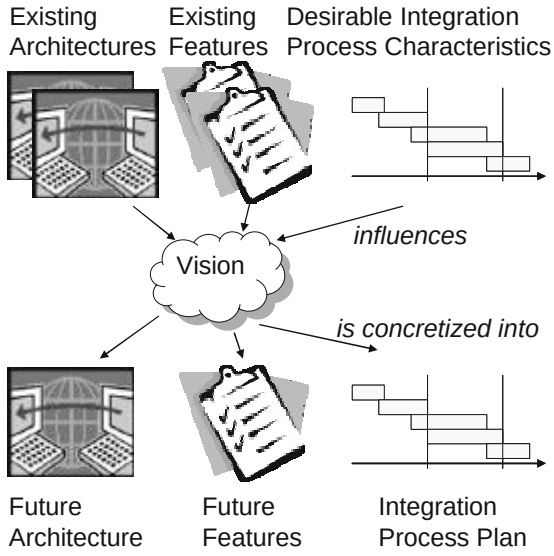

Figure 2: Interaction between decision and integration process

The specific questions addressed by this paper are:

Q1. Which are common experiences (good and bad) concerning vision concretization?

Q2. To what extent are the lessons learned from these experiences possible to generalize into recommendations for other organizations?

Section 2 describes related work, section 3 describes the methodology used in the research, and section 4 introduces the cases. Section 5 answers Q1 and Q2 by presenting experiences from the cases in the form of patterns, and section 6 concludes the paper by summarizing the most important observations and outlining future work.

2. Related Work

Three major fields of software integration are component-based software [8], open systems [6], and Enterprise Application Integration, EAI [1,7]. In a previous survey of existing approaches to software integration [3], we found that there is basically no existing literature that directly addresses the context of the present research: integration of software *completely controlled and owned within an organization*. To save space in the present paper, we refer to our previous survey of software integration for further references [3].

3. Research Methodology

To investigate these issues, we have carried out a multiple case study [9] with 9 cases from 6 organizations that have gone through such an integration process. Our main data source has been interviews, but in some cases we also had access to some documentation. In one case (F1) one of the authors (R.L.) also participated as an active member. Due to space limitations, we refer to a report accompanying the present paper were we have collected all raw data (primarily the copied out interview notes) [5]; the report also contains all details about how the research was carried out, including details about the cases and how research threats were addressed.

4. The Cases

Table 1 presents the cases very briefly; the report accompanying the present paper contains more details [5]. The cases are labeled A, B, etc. Cases E1, E2, F1, F2, and F3 occurred within the same organizations E and F. For the data sources, the acronyms used are I_X for interviews, D_X for documents, and P_X for participation, where X is the case name (as e.g. in I_A, the interview of case A), plus an optional lower case letter when several sources exist for a case (as e.g. for interview I_{Da}, one of the interviews for case D). I_X:n refers to the answer to question n in interview I_X. We have provided explicit pointers from the text into the report containing the data [5].

5. Analysis

By examining the data of the cases from the viewpoint of question Q1 asked initially, about common experiences, nine recurring patterns have been found. All of these patterns are based on several cases, which give confidence about the generality of the patterns, thus addressing Q2. The patterns are:

Small evaluation group. Statement: After higher management has identified some potential benefits with integration, a small group of experts should be assigned to evaluate the existing systems from many points of view and describe alternative high-level strategies for the integration. In cases C and F1 a small group evaluated the existing systems with the specific goal to identify how integration should or could be carried out, at the technical level (I_{Ca}:6, I_{Cb}:6,

Table 1: Summary of the cases.

	Organization	System Domain	Goal	Information Resources
A	Newly merged international company	Safety-critical systems with embedded software	New HMI* platform to be used for many products	*Interview:* project leader for "next generation" development project (I_A)
B	Organization within large international enterprise	Administration of stock keeping	Rationalizing two systems within corporation with similar purpose	*Interview:* experienced manager and developer (I_B)
C	Newly merged international company	Safety-critical systems with embedded software	Rationalizing two core products into one	*Interviews:* leader for a small group evaluating integration alternatives (I_{Ca}); main architect of one of the systems (I_{Cb})
D	Newly merged international company	Off-line management of power distribution systems	Reusing HMI* for Data-Intensive Server	*Interviews:* architects/developers (I_{Da}, I_{Db}).
E1	Cooperation defense research institute and industry	Off-line physics simulation	Creating next generation simulation models from today's	*Interview:* project leader and main interface developer (I_{E1}) *Document:* protocol from startup meeting (D_{E1})
E2	Different parts of Swedish defense	Off-line physics simulation	Possible rationali-zation of three simulation systems with similar purpose	*Interview:* project leader and developer (I_{E2}) *Documents:* evaluation of existing simulation systems (D_{E2a}); other documentation (D_{E2b}, D_{E2c}, D_{E2d}, D_{E2e}, D_{E2f})
F1	Newly merged international company	Managing off-line physics simulations	Possible rationali-zation by using one single system	*Participation:* 2002 (R.L.) (P_{F1a}); currently (R.L.) (P_{F1b}). *Interviews:* architects/developers (I_{F1a}, I_{F1b}); QA responsible (I_{F1c}) *Documentation:* research papers (D_{F1a}); project documentation (D_{F1b})
F2	Newly merged international company	Off-line physics simulation	Improving the current state at two sites	*Interviews:* software engineers (I_{F2a}, I_{F2b}, I_{F2f}); project manager (I_{F2c}); physics experts (I_{F2d}, I_{F2e})
F3	Newly merged international company	Software issue reporting	Possible rationali-zation by using one single system	*Interview:* project leader and main implementer (I_{F3}) *Documentation:* miscellaneous related (D_{F3a}, D_{F3b})

I_{F1c}:6, P_{F1a}, P_{F1b}, D_{E1a}). In case F1, users were also involved in this process, in order for them to grade different features of the existing systems (P_{F1a}, D_{F1a}). Important with this scheme is to involve both sides, as no single individual has overview of all systems (both cases concern newly merged companies). Also, everyone involved is partial and there is a clear risk that everyone "defends" their own system (I_{Cb}:6), there must be an open mind for other solutions than "ours" (I_{F3}:11). In the cases it appears that there has indeed been a good working climate with a "good will" from everyone (I_{Cb}:6, P_{F1a}). In both cases this was considered a good scheme; in case C the architects immediately saw that there were no major technical advantages of either system, and wanted to immediately discontinue one of the two systems, indifferent which, rather than trying the (I_{Cb}:6). The late decision (indeed, to discontinue one of the systems) was due to other reasons (see "timely decisions**"** below. A similar scheme was used in case E2, an external investigation was made, however with less technical expertise (I_{E2}:6, D_{E2a}).

Reusing experience from existing systems. Statement: *To be able to describe the envisioned system, one needs to understand the*

* HMI=Human-Machine Interface

state of the current systems (I_A:6, P_{F1a}, D_{F1a}, I_{F2e}:6, I_{F2f}:6, I_{F3}:11). Ideally, one would like to define the new system as consisting of the best parts of the existing systems; however, this is in practice not as simple as it first may seem. The requirements on the future system is clearly dependent on the experience of the previous systems, and can be stated in terms of existing systems (I_A:6, P_{F1a}, D_{F1a}, I_{F3}:6). However, this means that the requirements need not (some of the sources even say should not) be too detailed (I_A:5,6,11, I_{C1a}:6, P_{F1a}, D_{F1a}). In case A, the development organization explicitly asked sales people for "killing arguments" only, not a detailed list of requirements (I_A:5). This, combined with the experience and understanding of the existing systems, makes a detailed list of requirements superfluous (i.e. during these early activities; later a formal requirements specification may be required). The people developing the vision of the future system (e.g. a small evaluation group) need to study the other systems, preferably live (I_{Ca}:6, D_{E2a}, I_{F3}:6). Case F2 involves complex scientific physics calculations, and the study of the existing systems' documentation of the implemented models was an important activity (I_{F2e}:6, I_{F2f}:6). When looking at the state of the existing systems, an open mind for other solutions than the current way of doing things is essential (I_{F3}:11).

Improve the current state. Statement: *To gain acceptance, the efforts invested in the integrated system must not only present the same features as the existing system, but also improve the current state*. The existing systems must be taken into account (see pattern "reusing experience from existing systems"), but one should not be restricted of the current state (I_{F2f}:6); in case F2, it was indeed considered a mistake was to keep the old data format and adapt new development to it (I_{F2a}:9, I_{F2d}:7,9,11). The actual needs must be more important than to preserve the features of the existing systems (I_{F3}:11). One interviewee stated that a new system would take ~10 years to implement, and a merged (and improved) system must be allowed to take some years as well (I_{F2f}:6). In case E1, integrating several small, separate pieces as was envisioned required a more structured language (Ada), even though it would in principle be possible to reuse many existing parts as they were written in Fortran (I_{E1}:6); the organization was interested in Ada as such, which also contributed to this choice (I_{E1}:7).

Sufficient analysis. Statement: *Before committing to a vision, sufficient analysis must be made*. Obvious as that may seem, the difficulty is the tradeoff between the need of understanding the existing systems well enough without spending too much time. In case F2, insufficient analysis caused large problems: what was believed to involve only minor modifications resulted in complete re-design and implementation (I_{F2a}:9, I_{F2b}:9, I_{F2c}:3, I_{F2d}:6, 11). One method of ensuring sufficient analysis could be to use the "small evaluation group" pattern. Of course, pre-decision analysis somewhat contradicts the pattern "timely decisions"; a stricter separation from the actual integration process is also introduced, implying a more waterfall-like model which might not be suitable (I_{F1b}:5,6).

Strong project management. Statement: *To run integration efforts in parallel with other development efforts, a strong project management is needed* (e.g. I_{F1c}:9,11, I_{F2b}:5,11, I_{F2e}:9,11). To be able to control development, higher management and project management must have economical means of control (I_{Ca}:11, I_{F1b}:11). In case C, not until economical means of control were put into place did development of the system-to-be-discontinued stop (I_{Ca}:6). Case E1, a cooperation led by a research institute, can serve as a counter-example. Here, enthusiasm apparently was the driving force, and the lack of strict management was even pointed out as contributing to success (I_{E1}:9,11). Although we agree it is important to create a good and creative team spirit, we believe it would be a bad advice to recommend weak or informal project management, at least for larger projects.

Commitment. Statement: *It is not possible to succeed with integration if the efforts are half-hearted*. Commitment is needed from all stakeholders (I_{F1b}:11, I_{F1c}:11), which must also be accompanied with enough resources (I_{F1c}:11). In case F2 it was pointed out (based on negative experience) that for strategic work as integration is, one cannot assign just anyone with some of the required skills; the right (i.e. the best) people must be assigned, which is a task for project management (I_A:11, I_{F2b}:11, I_{F2d}:9,11, I_{F2e}:9,11). Realistic plans must be prepared, and resources assigned in line with

those plans (I_{F1c}:11). When directives and visions are not accompanied with resources, integration will be fundamentally questioned (I_{F1b}:3, I_{F1c}:6,9). When there is a lack of resources, short-term goals tend to occupy the mind of the people involved. Without a minimum effort in integration, the environment and the vision will change more rapidly than the integration makes progress, which means only a waste of resources. Integration will be doubted, which takes even more energy from the people involved. A long period of integration is problematic, since you need to maintain the existing system meanwhile (and for a while after they are retired as well) (I_{F2f}:6).

Make agreements and keep them. Statement: *To be able to manage and control a distributed organization formal agreements must be made and honored*. In case F2, it was pointed out as a big problem that requirements and design evolved driven by implementation (I_{F2b}:6, I_{F2c}:9, I_{F2d}:6, 11). Even in the informally managed case E1, the importance of agreeing on interface specifications and keeping them stable was emphasized (I_{E1}:7,9). More formalism than usual is required, you must have agreements written down and then stick to them (I_{F1c}:9,11).

Achieving momentum. Statement: *Achieving "momentum", i.e. an inner driving force is desirable*. (I_{F2f}:9) The external converging forces cannot be too strong for too long, which would take a lot of energy from the staff and the organization, will create stress and tension, and may also lead to a recurring questioning about the purpose of integration (I_{F1b}:3,11, I_{F1c}:6,9). One of the interviewees in case F1 (which has not made significant measurable progress during the 4 years that have passed since the company merger) asked "from where comes the driving force?" (I_{F1c}:9), pointing at the fact that integration is not a goal in itself. (These terms: converge, diverge, driving force, momentum, were terms used by many of the interviewees themselves).

Stepwise delivery. Typically, the vision lies far into the future, and integration processes are less predictable than other development projects (I_{F2c}:10,12). Maintaining the long-term focus without some way to monitor and measure progress is impossible (I_A:6,9, I_B:1, I_{Da}:12, I_{Db}:6, I_{F1b}:6, I_{F2c}:6,11, I_{F2f}:6). In contrast to development of new products, or new product versions, these activities are performed in parallel and often not considered the most important. For these reason the decisions regarding to the integration process do not only depend on the process itself, but also on many unrelated and unpredictable reasons. Stepwise deliveries and prototyping have been used for new development to increase process flexibility and this was also a recurring opinion among the interviewees. This could be one way of achieving the desirable momentum. There were some variations on this theme:

- Some of the interviewees maintained that there must be a focus on deliveries that gives user value, and a clearly identified customer (I_B:1,7,11,13, I_{F1b}:6,11). If it is possible to utilize a customer delivery to perform some of the integration activities, this will be the spark needed to raise the priority, mobilize resources, gaining commitment etc. (I_{F2c}:6,11). However, it should also be noted that customer delivery projects typically have higher priority than long-term goals such as integration, and may steal resources and commitment from the integration process. The extreme would be to focus only on immediate needs, questioning the need of integration at all (I_{F1b}:3,11, I_{F1c}:6,9).
- Case A used prototyping as a way to show an early proof of concept (I_A:1,6,9,11).
- In some cases where it has been difficult to formulate, or agree on, or commit to a vision, the opinion has been raised that you rather need to move on and do something more concrete. There might be too many unknowns, and the best way to carve out a more concrete vision is to do something that is useful in the shorter term, and use it as a learning experience (I_{F2c}:11, I_{F2f}:6). In case F2 requirements and design evolved uncontrolled, driven by implementation (I_{F2b}:6, I_{F2c}:9, I_{F2d}:6,11); better had been to either freeze the requirements or to include constant change into the development model.
- For a large system, a waterfall model is not suitable (I_{F1b}:5,6). It is often considered too risky to define the complete integrated system and implement it, as this runs the risk of not being feasible at time of delivery; there is a too long time to return of investment (I_B:1). Closely associated is the approach of a loosely integrated system: an integration point should be found and all

subsequent activities, although run as separate delivery projects, will little by little make integration happen (I_B:6,7, I_{F1b}:6,7,8,11; the proposed integration point in case F1 was a data storage format). There is however a tradeoff to be made, there are typically some common fundaments that need to be built (P_{F1a}, D_{F1a}, I_{F2e}:7).

- In order to develop and install a number of customer-specific systems in parallel, divergence can be allowed, if there are mechanisms that will enforce standardization and convergence from time to time (I_B:7,11,13).

Some interviewees proposed the opinion of not integrating at all. "Why integrate at all?" (I_{Cb}:7) is indeed a valid question, which will arise if a decision is not accompanied with priority and enough resources (I_{F1b}:3, I_{F1c}:6,9,11, P_{F1a}). Sometimes it might simply not be worth the effort to integrate – will the future savings through rationalization be larger than the integration efforts? (I_{F1c}:9, I_{F2d}:3). Reasonable project plans for reaching the vision must be considered; in case E2 there were very few resources available, which led to a very modest vision, in practice meaning no integration (I_{E2}:6).

6. Summary and Conclusions

We have studied 9 cases of software integration, and described recurring patterns for how to concretize the vision of the future integrated system. We asked two questions in the introduction; here these are repeated together with possible answers.

Q1. Which are common experiences (good and bad) concerning vision concretization? Answer: The patterns found are: *small evaluation group*; *reusing experience from existing systems*; *improve the current state*; *sufficient analysis*; *strong project management*; *commitment*; *make agreements and keep them*; *achieving momentum*; *stepwise delivery*. We also recapitulated voices from the cases questioning integration altogether.

Q2. To what extent are the lessons learned from these experiences possible to generalize into recommendations for other organizations? Answer: The fact that the same patterns replicated themselves across the heterogeneous systems and organizations of the cases gives some confidence about the generality of the results.

To conclude: only when a vision can be concretized into architecture, requirements, and an integration plan, and only when concrete milestones of the integration plan have been achieved is it possible to know that the vision is indeed an achievable vision and not a mirage.

6.1 Future Work

We are currently analyzing the same collected material from the three viewpoints presented: process, features, and architecture.

7. Acknowledgements

We want to thank all interviewees and their organizations for letting us take part of and publish their experience.

8. References

[1] Cummins F. A., *Enterprise Integration: An Architecture for Enterprise Application and Systems Integration*, ISBN 0471400106, John Wiley & Sons, 2002.

[2] Land R. and Crnkovic I., "Software Systems Integration and Architectural Analysis – A Case Study", In *Proceedings of International Conference on Software Maintenance (ICSM)*, IEEE, 2003.

[3] Land R. and Crnkovic I., "Existing Approaches to Software Integration – and a Challenge for the Future", In *Proceedings of Software Engineering Research and Practice in Sweden (SERPS)*, Linköping University, 2004.

[4] Land R., Crnkovic I., and Wallin C., "Integration of Software Systems - Process Challenges", In *Proceedings of Euromicro Conference*, 2003.

[5] Land R., Larsson S., and Crnkovic I., *Interviews on Software Integration*, report MRTC report ISSN 1404-3041 ISRN MDH-MRTC-177/2005-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, 2005.

[6] Meyers C. and Oberndorf P., *Managing Software Acquisition: Open Systems and COTS Products*, ISBN 0201704544, Addison-Wesley, 2001.

[7] Ruh W. A., Maginnis F. X., and Brown W. J., *Enterprise Application Integration*, A Wiley Tech Brief, ISBN 0471376418, John Wiley & Sons, 2000.

[8] Wallnau K. C., Hissam S. A., and Seacord R. C., *Building Systems from Commercial Components*, ISBN 0-201-70064-6, Addison-Wesley, 2001.

[9] Yin R. K., *Case Study Research : Design and Methods* (3rd edition), ISBN 0-7619-2553-8, Sage Publications, 2003.