# Measuring the Adoption of Software Processes

André Heijstek[1] and Hans van Vliet[2]
[1]SEI Europe
[2]Vrije Universiteit, Amsterdam, The Netherlands

## Abstract

*Often the work of a process working group is deemed ready when a new process is documented and placed somewhere on the intranet. Experience in process improvement however shows that at this point you have just passed milestone 1, and you still have 90% of the process improvement work ahead of you.*

*There is a target population out there still applying the old process(es). So you have to take actions to make them aware of what you have created, to make them fully understand what it takes to do a good job. Probably some pilots are needed, or people at least need to have the opportunity to try out your new process. And finally we need the whole organisation to apply the new process consistently, now and forever. This is the real daunting task of process improvement - an order of magnitude more complex than writing a process description. Here we don't work with paper and symbols that are patient and forgiving, we work with real people facing customer pressure, technical challenges, who often see 'us' process improvers as the bureaucrats, trying to make their life complicated.*

*The paper describes a measurement method that goes hand-in-hand with a structured approach to process adoption. This approach is based on a wealth of technology adoption and on many years of personal experience with process improvement and assessment. The notion of institutionalization or sustainability is especially emphasized. Too often processes are abandoned too soon after being introduced, so the measurement approach addresses this phase as well. The paper includes the results of a first validation of the approach.*

## 1. Short description / Stage setting

Many software process improvement (SPI) programs fail. Practitioners in the field of SPI claim that about 80% of those programs fail to achieve their promised benefits. Unfortunately there is no published data to support this statement, for an obvious reason: people do not publish failures. Studies show that 67% of SPI managers want guidance on *how* to implement SPI activities, rather than *which* SPI activities to implement [1], which can be taken as some evidence confirming the high failure rate claim. Even programs that do not fail as such often fail to deliver the promised returns. It appears to be very difficult to truly change the behavior of managers and engineers involved in software or systems development. Failure can be complete, where nothing of what was planned is achieved. But mostly, failure is much more subtle—some new processes are adopted to some extent, but not to the extent needed to achieve any real benefits. Improved processes do not bring any benefits if they aren't used well.

There are many consequences of failed adoptions:

- Organizations still do not achieve their development project goals, and projects often are too late, over budget, deliver lower quality than planned
- The investment in the improvement program is fully or partially wasted
- The improvement model (e.g., CMMI, Six Sigma, ISO 9000) is perceived by the affected organization as a bad model that does not deliver what is promised

It is difficult to analyze if improved processes have been fully adopted. For SPI programs based on the CMMI model [2] there is a well-defined appraisal method [3]. However, even with this rigorous method it is almost impossible to gauge if the processes have really changed the hearts and the minds of the practitioners.

This paper attempts to address these issues, focusing on understanding adoption to the level that we can influence and measure it. This paper first describes the research goals (section 2) and relevant literature (section 3), which are the basis for a validation approach (section 4). Section 5 describes the results of an actual implementation of the adoption measures and in section 6 we draw some conclusions.

## 2. Key Questions / challenges that we address

To support SPI programs to achieve better results one not only needs to introduce the right technologies and processes but these processes need to be adopted as well. Adoption is achieved by applying the right mechanisms [4] that support staff through technology adoption phases [5]. To track progress we need to be able to measure adoption. Besides, measuring adoption of processes will strongly support process compliance which is more and more mandated by external standards like Sarbanes-Oxley, ISO standards, etc.

### 2.1. Transition Mechanisms

It is widely recognized that process change or technology change occurs through several distinct stages [5], [6], [7, 8]. Prospective users of new processes first get in *contact* with the new process, become *aware* of what this process could do for them and obtain a deeper *understanding* of the technology itself. When these stages leave a positive impression users will perform a *trial usage* of the process before deciding on full *adoption*. The new processes become *institutionalized*, embedded in the culture, and after sufficient passage of time they become *internalized*; users cannot think they would ever perform the process in the old way again. Recognizing these stages, we could use them actively to support adoption in an SPI program. The design of the SPI program should identify and plan mechanisms to support the process users to proceed along these stages. This paper gives suggestions for identification of transition mechanisms and relates these mechanisms to adoption measurement.

### 2.2. Adoption Measurement

To make any founded claims about the (lack of) adoption of new processes, we should be able to measure adoption. To what extent do people apply a process? And how many people from our target population do apply the new processes? The following example illustrates the difference between partial and full adoption. An organization has developed a new estimation process to improve the accuracy of the development project estimates. The new process includes a spreadsheet which requires several attributes to be given (number of screens, number of interfaces, etc.) and then suggests an estimate for software development effort. The estimator is allowed to differ from this suggestion, but he then needs to document why.

A partial adoption of the new estimation process could have the following indicators:

- People use the new estimation spreadsheet.
- They compare the model outputs with what their gut-feeling estimate tells them.
- They then tune their inputs into the model to make both numbers match.
- When estimates turn out to be wrong, they blame the model.

A full adoption of the new estimation process might look like this:

- People use the new template and estimation model.
- They also compare model results with their gut-feeling estimates.

- They document where there are differences between the two and involve the estimation process owner in case of significant deviations.
- During project execution they compare their estimates to the actual numbers for effort and time for each of the tasks performed.
- Based on this the estimation model parameters are tuned to better match reality.

Key differences between real and partial adoption:

- The estimation method is used according to its intentions.
- The estimation results are integrated with the project tracking process. More generally speaking, the results of one new process are integrated with other processes, in ways that the process author may not have foreseen.
- The organization uses feedback to become a learning organization.

## 3. Review of Existing Literature

We review two groups of literature below. The first group describes theory on adoption of innovations in general. Several terms are used in literature: diffusion, transition, adoption, dissemination. In this article we use the term adoption in the description of our own work (but use the terms of the author when summarizing relevant articles). The second group of articles describes methods to measure adoption.

### 3.1. Adoption of Innovations

Rogers [6] has given an encyclopedic description of diffusion of innovations. He defines diffusion as the process by which an innovation is communicated through certain channels over time among the members of a social system. For this research this has two important implications: our transition mechanisms are the means to communicate innovations through certain channels, and the notion of 'members of a social system' is very important in defining the target group for adoption measurements.

Changes to processes should become changes to work practices, changes to behavior. Ajzen [9] describes in the theory of reasoned action (TRA) how intentions and beliefs may or may not lead to actual changed actions. According to TRA, a person's performance of a specified behavior is determined by his or her behavioral intention (BI) to perform the behavior, and BI is jointly determined by the person's attitude (A) and subjective norm (SN) concerning the behavior in question. As TRA describes behavior in a general sense, this model does also apply to behavior in the context of SPI. TRA is a descriptive model; it describes how behavior is driven by attitudes and subjective norms. We conjecture that absorptive capacity is missing in this model. TRA assumes that when one has the intention to adopt a certain behavior, this behavior will actually be performed. In reality people are typically constrained to adopt a new behavior by lack of time or lack of training.

### 3.2. Measurement of adoption

Zmud and Apple [10] describe a measurement method for technology infusion, using levels of use. They define infusion as: the extent to which the full potential of the innovation has been embedded within an organization's operational or managerial work systems. Examples of different levels of use are stand-alone, work flow, distributive system. The interconnectedness of tasks increases with infusion. The article describes a case study looking at diffusing a supermarket scanner. Some results of the study:

. innovativeness (measured as: earliness of adoption) is associated with diffusion.
. infusion is associated with both earliness and with diffusion.
. routinization seems to happen faster than infusion.

Zmud and Apple correctly make the case that this means that diffusion (or routinization) alone does not sufficiently describe adoption (or incorporation as they call it).
The method yields an ordinary infusion scale with three values: extensive use, integrative use, emergent use.

Saga and Zmud [11] identify three different facets of infusion: more extensive use of the innovation (e.g. using more technology features); more integrative use (using technology to create new workflow linkages among tasks); and emergent use (using technology to perform tasks not previously considered possible).

A model that has become quite popular to understand the problems in SPI programs comes from the family therapy domain [12], [13].

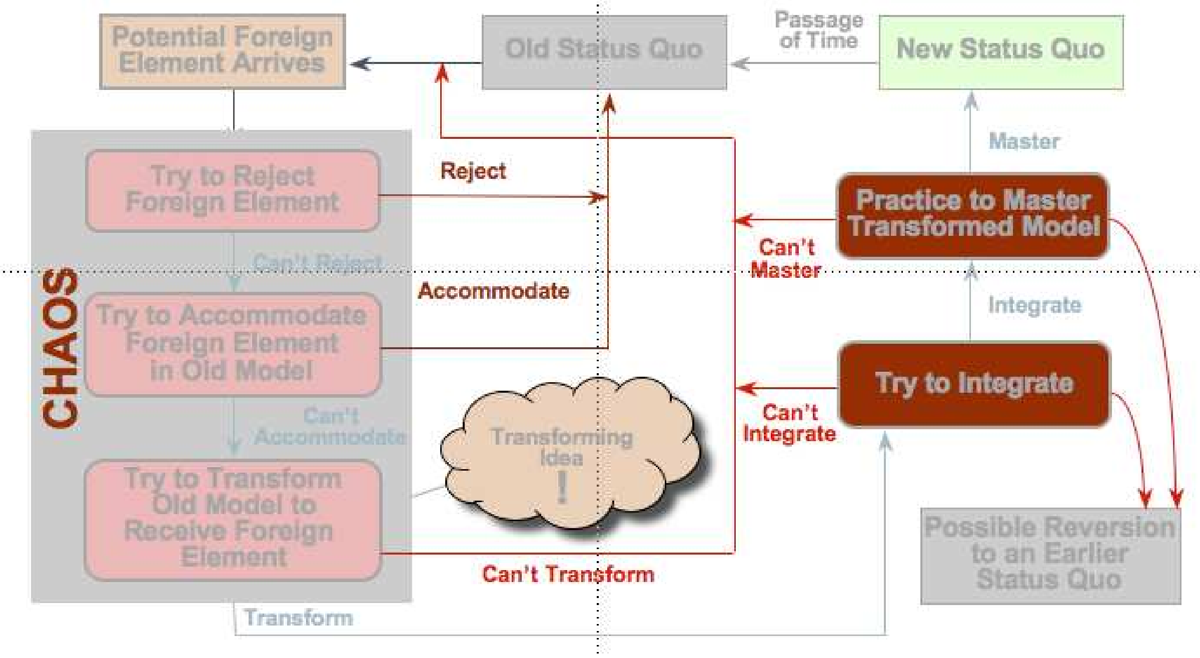



*Figure 1 – Satir's model of change [12]*

Satir describes a process of change as follows: we begin with the *status quo* of the organization (termed *old status quo* by Weinberg), a situation where people have learned to cope with the fallacies in the current system. Then a *foreign element* is introduced, in the context of SPI that would be the SPI initiative. The foreign element causes *chaos*, people who have long worked in the old status quo see all kinds of comfortable practices change, and the people and the system move into a state of disequilibrium. At some point in time a *transforming idea* is found that drives the system into *integration*, *practice*, and *new status quo*. The transforming idea from Satir is called transition mechanism in our approach.

The description here is sequential, however practice often is not. At each point in the process there is the possibility to fall back to the old status quo.

Weinberg [13] suggests that introducing new changes during the chaos phase should be avoided at all times, and mostly during the integration phase as well. Adding new changes during chaos and integration just increases the risk of falling back to old status quo. The optimum time for a new change is during the practice phase, shortly before new status quo. In that phase the chaos is over, so organizations are capable of adopting something new again, but not yet so fixed into a new status quo that new changes will lead to a major chaos again.

Satir shows that adoption can fail in many ways. Failure could be complete – the new process is not adopted at all – but also partly – the new process is just adopted to a limited extent. With our concept of Infusion we try to measure the extent of adoption.

## 4. Approach taken

Our overall research approach is the Action Research [14]. This paper represents one step in this approach, validating the adoption measurement concepts.

We created a theoretical model about software process improvement based on the literature review and our industrial experience, see Figure 2.

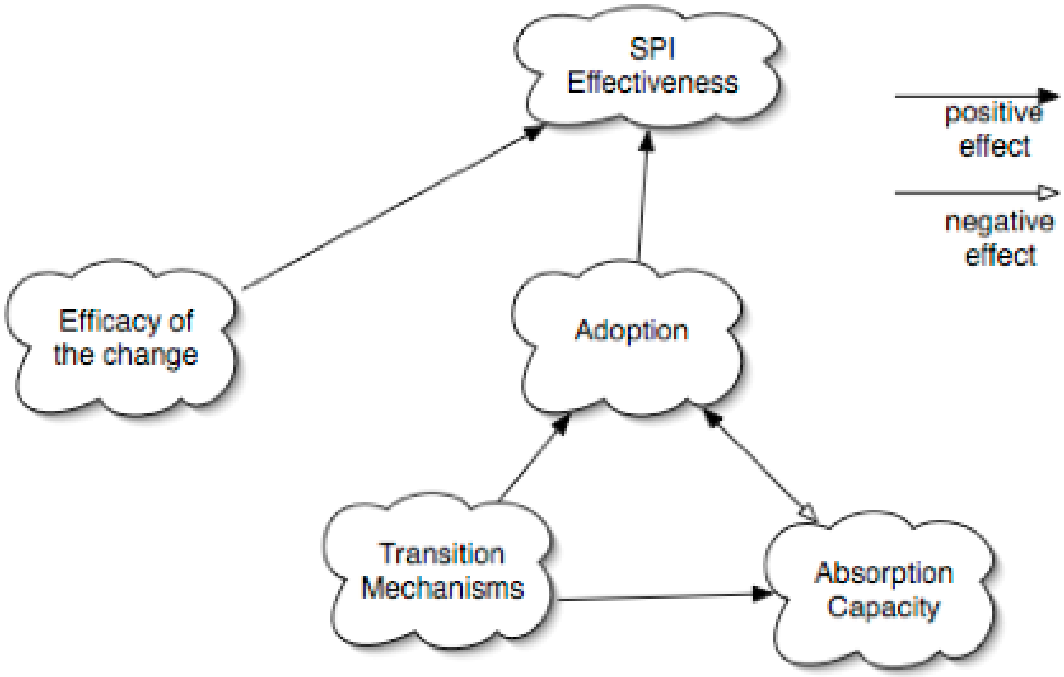



*Figure 2 - Modeling the effectiveness of SPI as a function of efficacy and adoption*

The model describes how the effectiveness of an SPI program depends on the efficacy of the change and the adoption of the change. Some authors combine efficacy and adoption into one concept; for example Wynekoop as quoted in [8] defines infusion as "the extent to which an innovation is used completely and effectively and improves the organization's performance". However, we claim that separating the orthogonal constructs efficacy–the content of the improvement program–and adoption–the method of change in the improvement program–gives us more analytic power. The efficacy of the change is a key successfactor. Have we implemented the right thing? Do the new processes solve the problems of the old situation? John Rost [15] gives some excellent examples of implementing the wrong set of processes in certain situations. For this study we assume that a useful approach has been taken. Our focus is on the adoption of the change. We contend that this depends on two main factors: the transition mechanisms used, and the absorption capacity of the organization. The concept of absorption capacity is described in [16] and will not be discussed in this paper.

### 4.1. Transition Mechanisms

Adoption of change typically follows a number of stages. Several models are described in the literature [6] [5]. In this article we apply the seven-phase adoption curve [5]. The phases are contact, awareness, understanding, trial use, adoption, institutionalization, and internalization. When organizations conduct a process improvement program, they should apply mechanisms to help the organization transition from one phase to the other. The more effective the mechanisms are, the more effectively and efficiently will the organizations adopt the new technology. There is not, and cannot be, any single set of transition mechanisms. Organizations have different backgrounds, cultures, and structures, so the set of mechanisms should be defined for each organization. Effective mechanisms from the past are of course a very good source for new efforts. Table 1 gives several examples of transition mechanisms for each phase, obtained from a customer engagement and from work by Garcia [17].

*Table 1. Transition Mechanisms*

| Stage | Transition Mechanisms |
|---|---|
| Contact | Information emails, Posters, Brown bag lunches |
| Awareness | "Elevator speech"<br>Magazine articles and conference briefings<br>Flash cards with objectives, benefits, URL, etc.<br>Web site, FAQ<br>Successful ROI stories, case studies |
| Understanding | Training sessions, communication<br>Detailed case studies<br>Identify and authorize champions<br>Identify stakeholder roles and responsibilities |
| Trial use | Coaching by process experts<br>Small working group to support pilots<br>Special authorities for pilots<br>Documented pilot results |
| Adoption | Availability of process artefacts<br>Strong set of incentives; rewards and consequences<br>Refined guidance on process usage choices<br>Education - mature courses, modularized for JIT delivery<br>In-Process Aids |
| Institutionalization | Inclusion in quality audits<br>Integration with other processes<br>Emergent use in unanticipated situations<br>Fully realized curriculum of training for different users<br>New employee training/orientation<br>Continuous improvement to adoption artifacts (guides, etc.) |
| Internalization | - |

## 4.2. Adoption Measurement

Process adoption is generally characterized as consisting of two orthogonal dimensions [17], [8]: infusion and diffusion. Infusion or depth of adoption describes how well the new process has been adopted by the target population. Is there a full or a partial adoption of the new processes? Diffusion or breadth of adoption describes how broadly the target population has adopted the new process. Do they all apply the new processes? It has been suggested by Kaputo and Garcia [4], [17] to use these dimensions as measurements for adoptions. An additional benefit of measuring adoption is that it provides a leading indicator for the return on investment (ROI) of process improvement. Assuming the right new processes have been developed, a high adoption measure tells us that a return on investment is to be expected.

*4.2.1. Measuring Diffusion*

To measure diffusion, we should measure how far the target population has come along the stages of the technology adoption curve [5]. When transition mechanisms have been defined in an improvement project, these mechanisms can be used as proxies for adoption commitment. Transition mechanisms are the means that have been used to get the target population through the stages of contact, awareness, understanding, trial use, adoption, institutionalization and internalization. Examples of transition mechanisms from some recent customer engagements can be found in **Table 2**.

***Table 2.*** *Suggested measurements for each stage in the adoption curve*

| Stage | Measurement |
|---|---|
| Contact | |
| Awareness | Number of attendees |
| Understanding | Number of attendees, course completion tests |
| Trial Use | Number of pilot projects, number of people reporting usage |
| Adoption | Number of projects/people reporting usage, practice implementation indicators, training attendees |
| Institutionalization | Process change requests |
| Internalization | |

To compute a composite diffusion measurement D, we count the number of people $P_S$ who are in a certain stage S of the adoption curve, and give weights to each stage $W_s$, internalization gets the highest weight, contact the lowest.

$$D = \frac{\sum Ps \times Ws}{n \times W \max} \qquad (1)$$

*4.2.2. Measuring Infusion*

Zmud only uses 3 levels for infusion: ordinary use, extensive use and emergent use. We propose to add 2 levels below ordinary use. Zmud's scale is sufficient for hard technologies, where the minimum level of use is just using the tool. However in soft technologies (or practice based technologies) there is a clear possibility to use a process below its intended level, to just pretend using it. So, we propose the following ordinal scale: no use, incomplete use, ordinary use, extensive use, emergent use.

To measure infusion we need to gauge how well the people within the target population use the new processes. The process developer should identify all roles that should work with this process, and for each role define how well the knowledge of the process should be and what the indicators are for each level of the scale. **Table 3** uses a configuration management process as an example.

***Table 3.*** *Using infusion facets as means to measure infusion*

| Role | Tasks | No use | Incomplete Use | Ordinary use | Extensive use | Emergent use |
|---|---|---|---|---|---|---|
| Developer | Version configuratio n items | | Only some items are maintained | All planned items are maintained | Upon check-in/out checks are | Structural information from the |

| | | | | | | |
|---|---|---|---|---|---|---|
| | (CI's), maintain integrity of a set of CI's | | in the repository (e.g. just code, no designs) | in the repository and given the correct labels | applied to ensure code quality | repository is used as input for product architecture updates |
| Tester | Retrieve code from the repository, version control test artefacts | | Not released code is retrieved to be tested. Test artefacts are not version controlled | Code with the correct status is retrieved, test cases are version controlled | Test reports refer exactly to which version of the code was tested with which version of the test cases | Info from the CM system is used to determine test intensity (e.g. many check-in/outs lead to more intensive tests) |
| Project Manager | Keep track of the status of CI's | | | Status info from the CM repository is an ordinary input to project tracking and reporting | | Info from the CM system is used to identify risks |

To compute a composite infusion measurement I, we count the number of people $P_s$ who are at a certain level L of infusion and give weights to each stage level of infusion, emergent use gets the highest weight, no use the lowest.

$$I = \frac{\sum Ps \times Ls}{n \times L\max} \quad (2)$$

*4.2.3. Presenting the measurement results*

An example of how an adoption measurement could be presented is given in Fig. 3. The X-axis shows diffusion, where the stages in the adoption commitment curve are numbered (0 = contact; 6 = internalization), the Y-axis shows infusion with 0 = not performed, 4 = emergent use. This first set of two graphs show a typical progress in adoption of a new process. First, in the left picture, most people are in the early phases of the adoption curve (contact, awareness, understanding), so at low levels of diffusion, and thus automtically at low levels of infusion. Later on, in the right picture, adoption has improved, more people are at later stages of diffusion, and start to apply the process in a good way, showing higher levels of infusion.

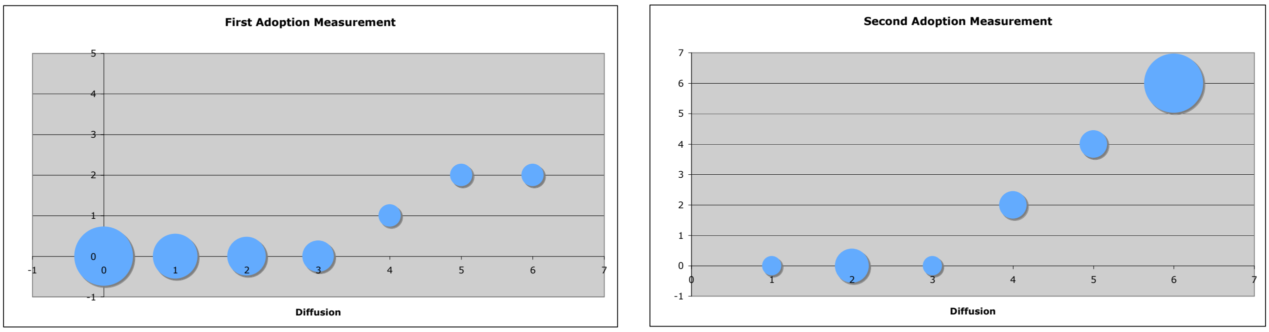
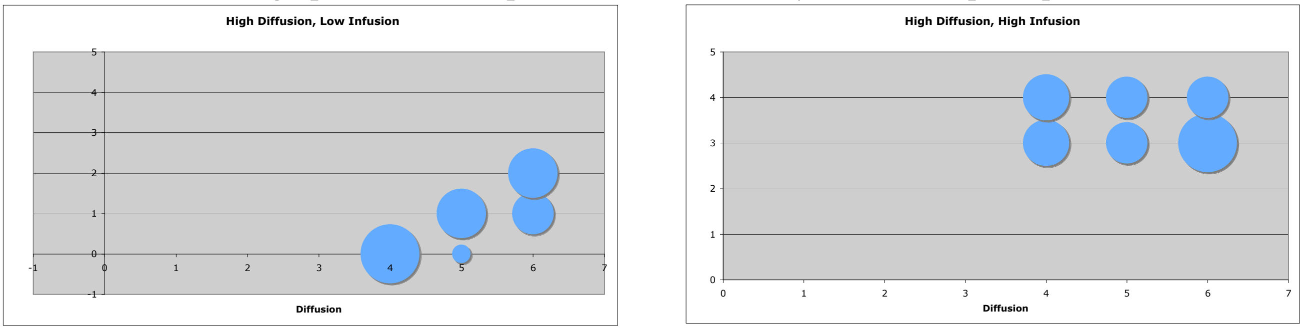



*Fig. 3. Graph of adoption as measured by diffusion and infusion*

The second set of graphs shows two processes with a clearly distinct adoption pattern.




The left graph show a situation with high diffusion and low infusion, so almost everybody applies the process, but almost nobody does it really well (so a positive return on investment is not to be expected). The right graph show a desired picture, high diffusion, high infusion: everybody does it well.

Note that the top-left part of the graph is an unreachable zone. High infusion is impossible to achieve without high diffusion.

## 5. Validation

The concepts discussed above have been validated in an industrial setting. The goals of the validation study were:

0 Show that these concepts can be applied in an industrial or business setting
1 Measure the time needed to perform these measurements (under the assumption that the method has no practical value if it is too costly to perform)
2 Comfirm the usefulness of the concepts in an evaluation with site personnel
3 Show that the measurement method yields actionable results
4 Show that the measure is valid

A software development department in a company producing optical equipment had introduced some improved configuration management processes supported by ClearCase (CC) as tool and several self-written build-tools. The department employs about 45 software development professionals. The target population for deployment was all developers (the project managers did not need to apply the process), totalling 40 persons. From this group a random sample of 10 persons was chosen. Seven of them were classified as ordinary users, three were called build liasons, performing somewhat advanced process tasks. The term build liason was not a formal role in the organization yet, it only existed in the minds of the process experts. But these experts knew which people did perform these advanced tasks.

### 5.1. Measurement Definition

We instantiated the diffusion measures in a meeting with the process improvement lead by eliciting the transition mechanisms that were used to deploy the process. From this set of mechanisms we derived indicators as described in Table 4.

*Table 4 – Diffusion Measures*

| Diffusion Stage | Transition Mechanisms |
|---|---|
| Contact | ▪ Software Colloquium<br>▪ Communication through steering group |
| Awareness | ▪ Usage survey |
| Understanding | ▪ Normal training course for developers<br>▪ Short training course for firmware people<br>▪ Inclusion of the normal training course in the software intro course |
| Trial Use | ▪ Usage of CC/BT was piloted first, then rolled-out<br>▪ Within the projects there is no possibility to try-out, however, the training course has exercises which allow trial use<br>▪ Archives from old projects were converted to CC and closed after a short period of double usage (double usage counts as trial use) |
| Adoption | ▪ CC was made mandatory to ensure uniformity with the US department<br>▪ Steering group created and enforced a roll-out plan<br>▪ Archives from old projects were converted to CC and closed after a short period of double usage (closing counts as adoption) |
| Institutionalization | ▪ Colloquium to keep people informed of changes<br>▪ Process descriptions are created and published on the intranet<br>▪ Correct usage of 'activities' was taught in the training<br>▪ Project managers enforce correct usage by checking their employees<br>▪ Incorrect 'activity' usage becomes visible when automatically generating release notes, thus enforcing correct use (albeit somewhat late)<br>▪ Every new employee is trained in CC/BT<br>▪ Usage Survey<br>▪ Metric shows number of failed builds |
| Internalization | |

The level of diffusion was measured using individual interviews with the sample group. The interview started with a very open question: "what did the organization do to get you to use ClearCase and Build tools", and was later followed up by more concrete questions like "have you also received any training?" if some indicators were not mentioned spontaneously.

The interviews were transcribed, and each interviewee was rated at a certain level of diffusion.

The effort needed to create these measures was:

*Table 5 – Effort to define measures*

| Activity | Effort for researcher (minutes) | Effort for organization (minutes) |
|---|---|---|
| Interview | 50 | 50 |
| Documentation | 50 | 10 |
| Review & Rework | 10 | 0 |

We instantiated the infusion measures in a meeting with the process experts for configuration management, asking the experts to give indicators that for them made the difference between the infusion scale elements. The results are documented in Table 6.

*Table 6 – Infusion measures*

| Infusion Level | Measurement Indicators for Ordinary Users | Measurement Indicators for Build Liasons |
|---|---|---|
| No Use | ▪ Considered impossible, without ClearCase there is no code available to work on. | |
| Incomplete Use | ▪ Unable to resolve a merge conflict without help of Build Liason<br>▪ Creating 'activities' at the wrong level of granularity.<br>Right:<br>- One activity per feature<br>- One activity per problem report, in other words: one activity per team track issue<br>Wrong:<br>- One activity for the whole project<br>- One activity per detailed task<br>▪ Incorrect use of Rebase/Deliver. Correct use is to Rebase frequently, every 1-2 days. Before Delivering a quick test should be performed ensuring correct local functioning. Rule-of-thumb: the longer the time after the Rebase, the more intensive the test should be. | ▪ Incorrect project setup. Correct procedure is to propose a setup to the CM experts and get their approval.<br>▪ No pro-active checking of the build results – indicated by a subscription to the automatic build results email. |
| Complete Use | ▪ Regular check-out/in of source code<br>▪ Regular Rebase/Deliver activities<br>▪ Resolving merge problems | ▪ Helping developers to resolve merge issues |
| Extensive Use | ▪ Performing Build Liason tasks<br>▪ Initiating new ideas on better usage of CC/BT (indicated by issuing Change Requests)<br>▪ Creating customization scripts within the BT<br>▪ Retrieving reports from CC<br>▪ Submitting Problem Reports | ▪ Working with multi-project issues: Right: resolving a bug once and using CC functions to propagate the solution to other branches<br>▪ Wrong: making the same edits twice in both branches (rate as Incomplete Use) |
| Emergent Use | ▪ Submitting Change Requests | |

The effort needed to create these measures was:

*Table 7 – Effort to collect measures*

| Activity | Effort for researcher | Effort for organization |
|---|---|---|
| Interview | 4:18 hours for 10 interviews | 4:18 hours |
| Documentation | 45 minutes to score all 10 interviews<br>20 hours to transcribe the interviews | No |

## 5.2. Measurement Data Collection

With these measurement definition, we then interviewed a random sample of 10 developers. This resulted in the following adoption graph.

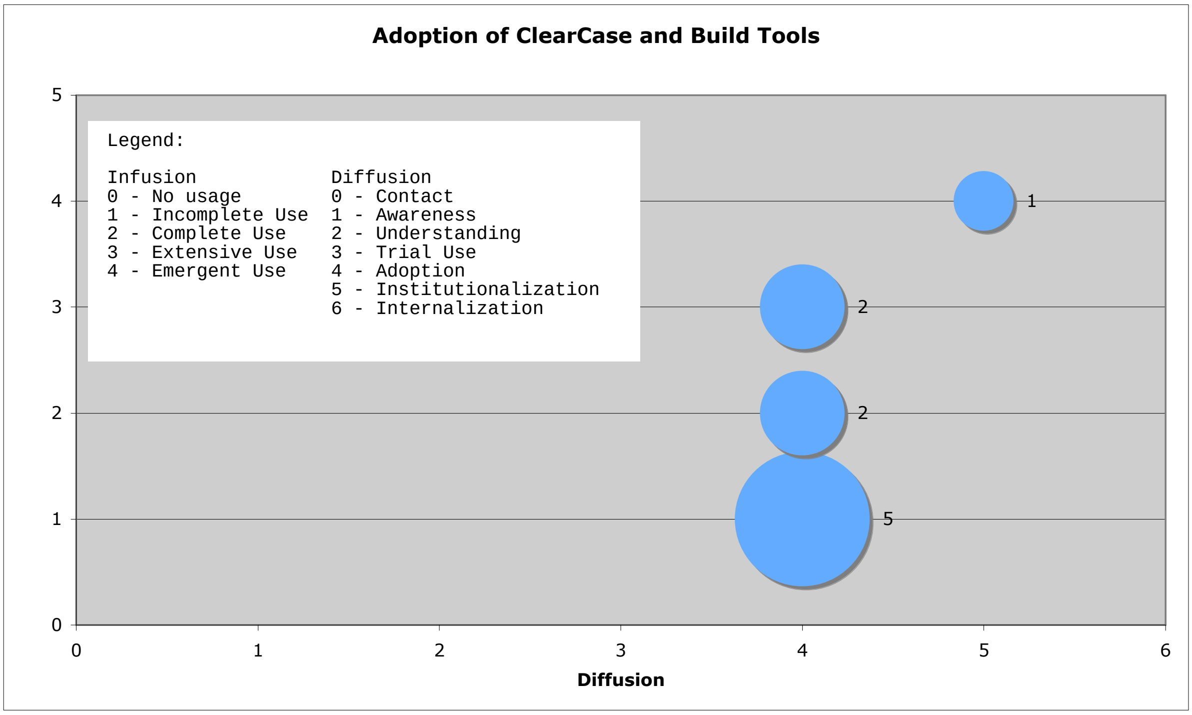



*Figure 3 - Adoption of ClearCase and Build Tools*

So we can see that the process is well diffused (see formula 1: diffusion = 0.73), there is nobody not using it on a daily basis. However in general the institutionalization stage is not yet achieved. This is caused by a lack of management follow-up and auditing of correct usage.

The level of Infusion is relatively low (see formula 2: infusion = 0.38). We defined 3 criteria for Incomplete Use:

- ability to perform merging;
- defining activities at the right level;
- rebasing before delivery

Of these, the rebasing was never a problem (this was enforced by the tools), but many users indicated they had trouble with merging, especially the user interface of the merge tool was perceived as confusing, so either tool improvements or training is needed to resolve this problem. Activities were also often ill-defined. Some users created just one activity for the whole project, several users had problems with the definition of a 'feature'.

Very few users tried to go beyond ordinary use of the technology. The few who had issued change requests for improvements had noticed either rejection or long response times, and have since then given up.

## 6. Conclusions

The study shows that it is possible to define and collect adoption measures with very limited effort. The results as presented give the organization a clear picture of where they stand with respect to adoption, and the reasons for not achieving higher levels of infusion and diffusion are concrete enough to allow focused improvement actions.

A corollory effect of defining the measurements is an increased awareness with the process experts for the need of transition mechanisms. During our definition session, one of the experts stated: "with every issue discussed as incomplete use, I realized: we have not communicated this well enough to our staff".

In future research we intend to compare the interview style measurement with a questionnaire based measurement, using a Guttman scale as basis. Expected advantages of this approach are even lower cost of data collection and more repeatable measurements – there is less interpretation needed to score infusion and diffusion. A potential disadvantage is that we loose insight into why a higher level of adoption is not achieved, thereby losing the ability to quickly define improvement actions.

## 7. References


[1] J. D. Herbsleb and D. R. Goldenson, "A systematic survey of CMM experience and results." in 18th International Conference on Software Engineering conference proceedings, 1996.

[2] SEI, "Capability Maturity Model Integration (CMMI), Version 1.1", Addison Wesley, 2002.

[3] SEI, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document", Publisher, 2001.

[4] K. Caputo, "CMM Implementation Guide: Choreographing Software Process Improvement", Addison-Wesley Professional, 1998.

[5] D. Conner, R. and R. Patterson, W., "Building Commitment to Organizational Change", Training and Development Journal, 1982, pp. 18-30.

[6] E. M. Rogers, "Diffusion of Innovations", The Free Press, 1995.

[7] M. J. Gallivan, "Strategies for Implementing new Software Processes: an Evaluation of a Contingency Framework", SIGCPR/SIGMIS, 1996, pp. 313-325.

[8] M. J. Gallivan, "Organizational Adoption and Assimilation of Complex Technological Innovations: Development and Application of a New Framework", Database for Advances in Information Systems, 2001, pp. 51-85.

[9] I. Ajzen and M. Fishbein, "Unterstanding Attitudes and Predicting Social Behavior", Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1980.

[10] R. W. Zmud and L. E. Apple, "Measuring Technology Incorporation / Infusion", Journal of Product Innovation Management, 1992, pp. 148-155.

[11] V. L. Saga and R. W. Zmud, "The Nature and Determinants of IT Acceptance, Routinization, and Infusion", Diffusion, transfer, and implementation of information technology: proceedings of the IFIP TC8 Working Conference on Diffusion, Transfer and Implementation of Information Technology, Pittsburgh, PA. U.S.A., 11-13 October 1993, 1994, pp.

[12] V. Satir, J. Banmen, J. Gerber and M. Gomori, "The Satir Model - Family Therapy and Beyond", Science and Behavior Books, Inc., Palo Alto, CA, 1991.

[13] G. Weinberg, "Quality Software Management: Anticipating Change", Dorset House, 1997.

[14] D. Remenyi. et al., "Doing Research in Business and Management", Sage Publications, 2000.

[15] J. Rost, "Software Engineering Theory in Practice", IEEE Software, 2005, pp. 94-95.

[16] A. Heijstek and H. Vliet, van, "Less is More in Software Process Improvement", in conference proceedings, 2005.

[17] S. Garcia, "Are you prepared for CMMI?" Crosstalk, 2002, pp. 19-23.