Patterns for Conducting Process Improvement

Brad Appleton *<bradapp@enteract.com>*
http://www.enteract.com/~bradapp/

Chicago Software Process Improvement Network
(C-SPIN)

February 11, 1998

1. Who am I & Why am I Here?

Who I am:

- Brad Appleton, Senior Software Engineer, Motorola AIEG
- Practicing software developer since 1987
- Work primarily on software development tools
- Special emphasis on:
  - Software Configuration Management (SCM)
  - Object-Orientation (O-O)

Why I'm here:

- Surviving practitioner of several improvement efforts
- Noticed some successfully recurring "best practices"
- Researched them, and wrote them up

2. Abstract

- Process Improvement and Product Development have many things in common
- Recognizing these similarities is important
- So is recognizing the differences
- Process change entails cultural change
- Numerous social & technical barriers to overcome

There are some recurringly successful strategies used to address many of the above!

3. Introduction & Acronyms

SPI: Software Process Improvement

PIT: Process Improvement Team (a.k.a. PWG, SEPG)

PEG: Process Engineering Group (a.k.a. SEPG)

IAT: Improvement Action Team

My experience in roughly a half-dozen SPI efforts:

- Served as both a change-agent and a change-target
- Both Software CMM and ISO-9000 focused SPI efforts
- Some successful strategies were common to most of them
- Published SPI experience reports described many of the same strategies
- Documented these recurring “best practices” as “*patterns*”

4. What are Patterns?

Trendy: Recent "hot topic", OOD buzzword, lots of hype!

Literary: Form of software engineering problem-solving documentation

Pragmatic: Describe practical solutions to "real world" problems

Recurring: Identify good design structures which recur in practice

Generative: Show how and when to apply the solution, and generate the desired design structure

Emergent: Larger solutions emerge indirectly from applying patterns in succession, and in concert together

5. Pattern Origins and History

- Writings of architect Christopher Alexander
  (coined this particular use of the term *"pattern"* ca. 1977-1979)
- Documentation of best practices and handbooks for engineering and architecture
- Literate programming (Don Knuth), ca. 1984
- Kent Beck and Ward Cunningham, Tektronix, OOPSLA'87
  (used Alexander's *"pattern"* ideas for Smalltalk GUI design)
- Erich Gamma, Ph.D. thesis, 1988-1991
- James Coplien, Advanced C++ Idioms Book, 1989-1991
- Gamma, Helm, Johnson, Vlissides, (*"Gang of Four"*) Object-Oriented **Design Patterns** book, 1991-1994
- PLoP Conferences and books, 1994-present

6. Pattern Definitions

A “pattern” is ...

- An abstraction from a concrete form which keeps recurring in specific, non-arbitrary contexts *[generic definition]*
- A recurring solution to a common problem in a given context and system of forces *[Alexander]*
- A named “nugget” of instructive insight, conveying the essence of a proven solution to a recurring problem in a given context amidst competing concerns
- A successfully recurring “best practice” that has proven itself in the “trenches”
- A literary format for capturing the wisdom and experience of expert designers, and communicating it to novices

7. Kinds of Software Patterns

- Design Patterns (software design; often object-oriented):
  - architecture (systems design)
  - design (component interactions)
  - programming idioms (language-specific techniques/style)
- Analysis Patterns (recurring & reusable analysis models)
- Organization Patterns (structure of organizations/projects)
- Process Patterns (software process design)
- Domain-Specific: *Any other domain you can think of!*

8. The Problem of Process Improvement

- Process improvement affects *more* than just the process
- Process improvement efforts disrupt delicate ecosystems deeply rooted within the community
- Process change means culture change
- Culture change entails changing the perceptions, values, and normative behaviors of a community
- Requires buy-in/participation from everyone affected:
  - Senior Management
  - Middle Management
  - Program/Project/Product "Line" Management
  - Practitioners & SQA

9. Opposing Forces of SPI

- Resistance to change; perceived threat of losing:
  - power
  - control
  - familiarity
  - social/professional status
- Speed at which groups/individuals can assimilate change
- Organizational climate's tolerance/readiness for change
- Process change imposes a learning curve
  (things appear to get worse before they get better)
- Improvement efforts consume time and resources
  (which many would prefer to spend on current development projects)

10. Patterns for Conducting SPI

Organization Patterns	Process & Communication Patterns
• Local Heroes	• Process is Product
• PIT also Practices	• Virtual Forum
• Dedicated Improvement Processors	• Process follows Practice
• Center PEG	• Improvement follows Process
• Improvement Action Teams	• Improvement follows Spiral

- These patterns are *not a complete set of solutions* for conducting SPI
- Their repeated *success has been documented* throughout the published SPI literature
- Many *issues left unresolved/unaddressed* are discussed within their resulting contexts

11. Applicability of these SPI Patterns

These patterns seem to be applicable when:

- Senior management commitment has been obtained
  - This is a hard problem all by itself, but is not addressed by these patterns
- Process goals/assessment criteria have been determined
  - Typically one of: ISO 9000, the SEI Software CMM, or SPICE

The circumstances of my own personal SPI experiences:

- Size of the groups ranged from 7-70 people
- Encompassed 1-10 project teams within the group
- Project team sizes were between 2 and 12 people
- Typically considered small-medium sized SPI efforts
- Published SPI case studies suggest these patterns scale to larger groups (perhaps with variations)

12. Pattern Elements

Name

- a meaningful “conceptual handle” for discussion

Context

- tells *how the problem occurs* / *when the solution works*

Problem

- statement of the problem / *intent* of the solution

Forces

- trade-offs, goals+constraints, motivating factors/concerns
- tells *why the problem is difficult*

Solution

- tells *how to generate* the solution
- the solution structure, its participants & collaborations

13. Pattern Elements *(cont.)*

Examples *(optional)*

Resulting Context

- describes the end result, benefits and consequences
- shows how the forces were balanced/traded-off
- tells *how the solution works out*

Rationale *(optional)*

- underlying principles/heuristics justifying the solution
- tells underpinnings of *why the solution works out*

Related Patterns

- patterns which are similar, or may precede/follow this one

Known Uses

- 3 or more independent instances of “real world” success

14. Process is Product

Context:

- Senior management has committed to support SPI efforts
- You are responsible for mobilizing people and resources to make it happen
- SPI is a new endeavor for your group
- You're unsure how to get started and get organized

Problem:

How should a process improvement initiative be organized and managed?

Forces:

- Want to use a familiar/established project management infrastructure
- You are trying to change the process, not develop a software product
- What works for a development project may not work for SPI
- Unless treated like other projects, SPI may not get needed consideration (and respect) from practitioners and from upper management

15. Process is Product *(cont.)*

Solution:

- Treat it like a development project!
- Recruit a corresponding project team (PIT) and project leader
- Establish a repository for process documentation and other process artifacts
- Use appropriate planning, tracking, configuration management, etc.
- Ensure visibility of the SPI project to upper management and the rest of the organization is comparable to that of other important projects

Resulting Context:

- Familiarity: the project management infrastructure is well established
- Uniformity: common project management framework for process & products
- Visibility: assists in obtaining management resources and support
- Credibility: Helps legitimize SPI efforts so they are taken seriously

Known Uses:

- Kodak *[Wiegers]*; Hewlett-Packard *[Grady]*; NORAD *[Wakulczyk]*; Lloyds Bank *[Larner]*; Several SPI case studies *[Radice], [Austin,Paulish] & [Curtis]*

16. Virtual Forum

Context:

- Setting up the SPI project infrastructure
- The PIT needs to regularly communicate with process stakeholders to announce project status, and to solicit feedback and participation

Problem:

How do you periodically discuss improvement efforts without numerous group-wide meetings that interrupt/compete with product development?

Forces:

- Want to keep all stakeholders informed of SPI efforts/status/progress
- Want practitioner input/feedback on SPI (since they have to live with it)
- Coordinating schedules to accommodate everyone can be a nightmare
- Widespread sentiment that meetings detract from accomplishing “real work”

17. Virtual Forum *(cont.)*

Solution:

- Create a group-wide discussion forum using a two-way communications medium already in wide use (local newsgroups, intranet, notes, etc.)
- Make sure messages on the forum are automatically archived/backed-up
- Announce its availability and encourage its use for SPI input+feedback
- Establish guidelines/policy for proper use (netiquette)

Resulting Context:

- Can communicate many SPI issues without having to coordinate schedules
- Face-to-face meetings still needed, but with reduced frequency
- Enables high-frequency 2-way communication between PIT and its customers
- Human contact is not replaced, but augmented by technology
- Forum archive serves as SPI project “memory”, preserving important historical comments that may be easily recalled

Known Uses:

- SPI experience reports and case studies *[Austin,Paulish], [Baumert], and [McLane]*

18. Local Heroes

Context:

- Need to assemble the process improvement team (PIT)
- Need to consider people both external and internal with varying experience

Problem:

How do you staff the PIT with members who can effectively lead the practitioner community in accepting and adopting process changes?

Forces:

- Process experts often perceived as steeped in theory rather than practice
- Want people with process knowledge and expertise
- Want people with solid, real-world experience in the trenches
- Internal people know the current climate, but may be less versed in SPI
- Outsiders might be experts, but aren't part of the community
- Trust/respect of key practitioners is needed to gain inroads into the community

19. Local Heroes *(cont.)*

Solution:

- Use "all-stars in the family": respected members of the organization with proven track records as developers or managers
- Try for equitable representation from the various projects
- But do *not* sacrifice experience and respect in order to achieve the above
- If you have to compromise, go with the more influential individuals

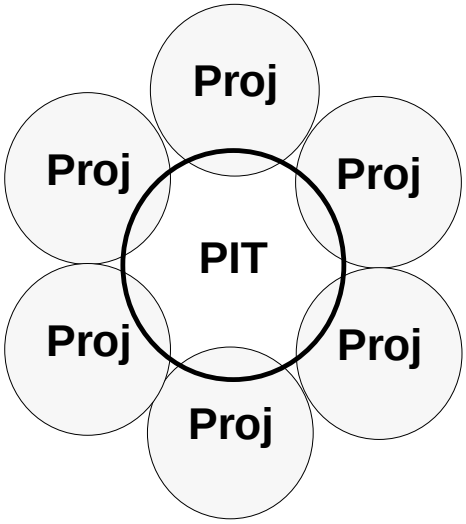
Resulting Context:

- The PIT is both socially and technically aligned with the practitioner community
- PIT members have intimate knowledge of development issues and people (and their deeds and words are respected within the development culture)

Known Uses:

- NORAD *[Wakulczyk]*; SPI experience reports and case studies *[Curtis], [Fowler,Rifkin] and [Donaldsen,Siegel]*

20. Local Heroes *(cont.)*



21. Center PEG

Context:

- SPI for a large group
- One PIT will either be too big/unwieldy, or won't be enough
- Need multiple PITs

Problem:

How do you organize and manage multiple PITs for a large-ish group?

Forces:

- A single guiding coalition is good for conceptual integrity and consistency
- One PIT with equitable representation will be too big and unmanageable
- Several smaller PITs require significant extra effort for coordination and communication
- Issues of authority and control may arise between the various PITs

22. Center PEG *(cont.)*

Solution:

- Create a Process Engineering Group (PEG) to be a center of guidance and support for the other PITs (similar to a SEPG in the Software CMM)
- PEG members will typically work full-time on process engineering and improvement (see Dedicated Improvement Processors)
- PEG is the primary authority for conducting/organizing SPI in the organization
- Variant #1 - PIT per Subgroup: local PITs address the entire software process for their own subgroup or department
- Variant #2 - PIT per Core Competency: each local PIT focuses on a single KPA, using domain experts from across the organization

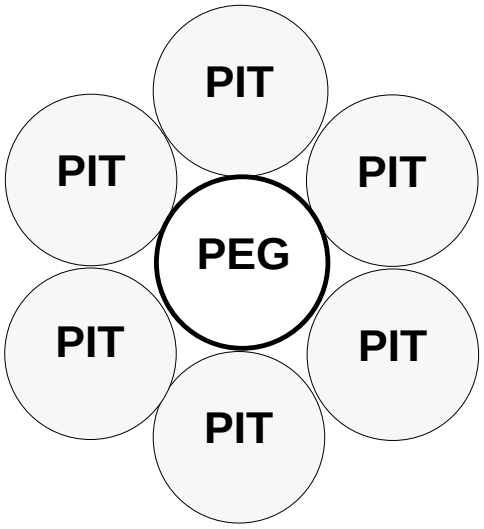
Resulting Context:

- PEG becomes a central hub of SPI coordination, communication & guidance
- The PEG and PITs are typically used throughout the life-span of SPI

Known Uses:

- Raytheon *[Haley]*; GTE *[Dorsey,McDonald]*; DuPont *[Austin,Paulish]*; *[Donaldsen,Siegel]*

23. Center PEG *(cont.)*



24. PIT also Practices

Context:

- Need to estimate and request SPI resources (including people and effort)
- The time/effort requested of each person may determine whether or not they can participate in the PIT

Problem:

How much time should PIT members devote to SPI to make reasonable progress without becoming detached from the practitioner community?

Forces:

- Part-time may not be enough to contribute the necessary time/resources
- Nice to have people who can dedicate the majority of their time to SPI efforts
- Some Local Heroes are too important to current projects to be pulled off
- Small groups may not be able to spare any single person for full-time SPI
- Current practitioners are desirable because they're intimately aware of existing corporate culture and the practitioner community

25. PIT also Practices *(cont.)*

Solution:

- Have PIT members spend 10%-20% of their time on SPI while still working on their current development projects.
- Make sure their workloads are adjusted to permit time for SPI activities (This requires management cooperation and support)
- Try to have at one or two PIT members devote 50%-100% of their time to handle managerial and administrative overhead for coordinating SPI efforts
- Is it realistic to expect to accomplish SPI with a part-time team?
  - 4-5 hours per week per person isn't very much, especially if PIT meetings are held on a periodic basis (e.g., weekly or bi-weekly)
  - 8-10 hours per week per person is more realistic, *provided that* workloads can be adjusted accordingly

Resulting Context:

- The PIT remains socially connected with the practitioner culture/community
- PIT members may not work full-time on SPI, but maintaining this connection greatly facilitates process changes being adopted and accepted

26. PIT also Practices *(cont.)*

- *Risk*: part-time SPI efforts may disappear whenever a crisis arises!
  - This would jeopardize the continuity and conceptual integrity of SPI efforts
  - Partially addressed by devoting 1-2 people half-time or full-time to SPI

Rationale:

- Some warn against committing people only part-time to SPI efforts
  - "No pain! No gain!": taking a "hit" early on will pay off in the long run
  - But many groups (especially small ones) simply can't afford the initial investment (if the "early hit" kills you, you won't be around for "the long run")
- Nothing wrong with taking "baby steps" if that's all you can presently spare
- Things may take longer to accomplish, and one still needs to worry about improvement efforts dwindling in a crunch
- But, better to proceed slowly and reach the goal than overcommit and fail
- Previous failure will make it doubly difficult to try again

Known Uses:

- NORAD *[Wakulczyk]*; Kodak *[Wiegers]*; GTE *[Dorsey,McDonald]*; DEC *[McLane]*

27. Dedicated Improvement Processors

Context/Problem/Forces: *(see PIT also Practices)*

Solution:

- Have PIT members dedicate their efforts full-time to SPI
- PIT members regularly spend time assisting projects in performing the process
- Thus, in addition to conducting SPI efforts, PIT members serve as hands-on mentors to assist performing and tailoring the process for the other projects

Resulting Context:

- PIT has ample time and resources; SPI need not progress at a snail's pace
- Conceptual integrity and continuity of SPI is less at risk with full-time personnel
- Opts for the opposite extreme from PIT also Practices: members are more isolated from the development teams, but devote more time to effect SPI
- Tries to manage this risk (greater isolation) by having PIT members regularly interact with the development project teams.

Known Uses:

- Bull HN *[Herbsleb,Carleton]*; SPI case studies *[Fowler,Rifkin] and [Donaldsen,Siegel]*

28. Process follows Practice

Context:

- SPI project infrastructure has been set-up and a PIT has been assembled
- Need to commence trying to change/adapt the process to meet SPI goals

Problem:

How do you change the process to meet SPI goals while ensuring the process documentation accurately reflects what really happens in the trenches?

Forces:

- The desire to begin making process changes right away can be very strong
- So can the need to demonstrate visible progress ASAP (to gain confidence and credibility in SPI efforts from managers and practitioners)
- This flies directly in the face of: resistance to change, speed and size of change (evolution versus revolution), and tolerance for change
- Want to change process documentation to address the assessment criteria
- Also want documented process to be genuinely used and followed (as opposed to shelfware that simply stays on the shelf)

29. Process follows Practice *(cont.)*

Solution:

Start by discovering and understanding current practice throughout the group. Then iteratively and incrementally improve the process and ensure that documentation is updated appropriately

1.Cherchéz les Documentation! (Archaeology)

- Find any existing process documentation (excavate process artifacts)

2.Know Thyself! (Anthropology)

- Talk to practitioners to discover current practices, and understand how work tasks are performed
- Reconcile differences between actual and espoused processes

3.Process follows Practice! (Characterize)

- Document these current practices, bringing together all artifacts
- Then review and baseline the result!

4.Piecemeal Growth! (Incremental/Iterative Improvement)

- Assess current versus desired state and identify possible improvements
- Implement and evaluate improvements, deploy what works

30. Process follows Practice *(cont.)*

Resulting Context:

- The first three activities form a lifecycle model for *process definition*
- The last activity outlines a basic lifecycle structure for *process evolution*
- Evolutionary/incremental approach balances resistance/tolerance/speed:
  - Improvement progress is slow during archaeology and anthropology phases
  - Necessary to analyze/understand stakeholders and assess change impact
  - "If you don't know where you are, a map won't help!" -- Watts Humphrey

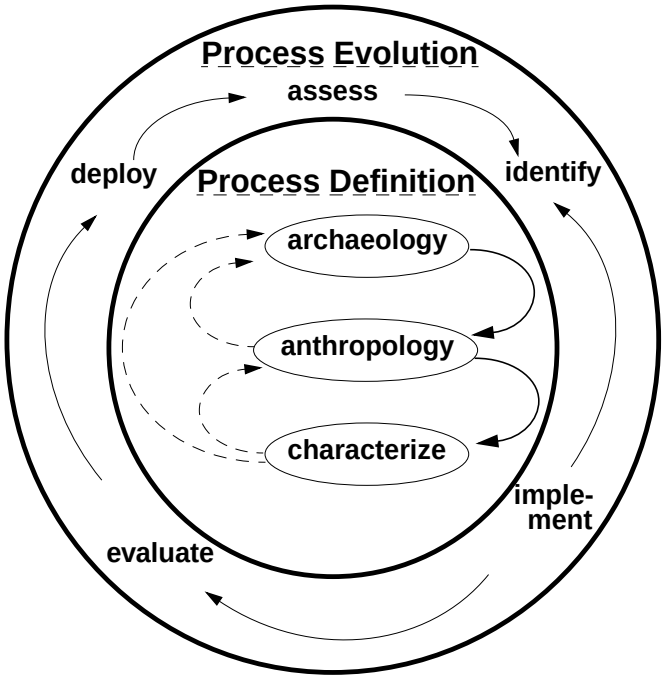
Rationale:

- Builds on Process is Product by saying SPI is legacy systems reengineering
- Proposing *complete* overhaul sends a message: you're doing *everything wrong*
- Many things may need improvement, but many things are being done *right*
- Process follows Practice makes clear what *does* and *does not* need to change!
- Increases familiarity & self-esteem; decreases the size and speed of change

Known Uses:

- From SPI experience reports *[Krasner], [Austin,Paulish], [Fowler,Rifkin], and [WeinbergV4]*

31. Process follows Practice *(cont.)*



32. Improvement Action Teams

Context:

- A specific process area has been selected for improvement
- Some preliminary planning and discussion have already been conducted

Problem:

To facilitate its acceptance while making effective use of time and effort, who should implement and deploy a given improvement idea?

Forces:

- PIT (or PEG) is primarily responsible for leading process improvement efforts
- Process changes are most likely to be accepted when developed in participation with their practitioners
- PIT has been granted time and resources for SPI
- This may not be true of remaining practitioners outside the PIT

33. Improvement Action Teams *(cont.)*

Solution:

- Form an Improvement Action Team (IAT) from the pool of PIT members and practitioners who championed or supported the improvement idea
- The IAT should be *small*, and *tightly focused* on the *single improvement*
- Non-PIT members should devote 10%-20% of their time to the improvement
- Disband the IAT after the improvement has been successfully deployed

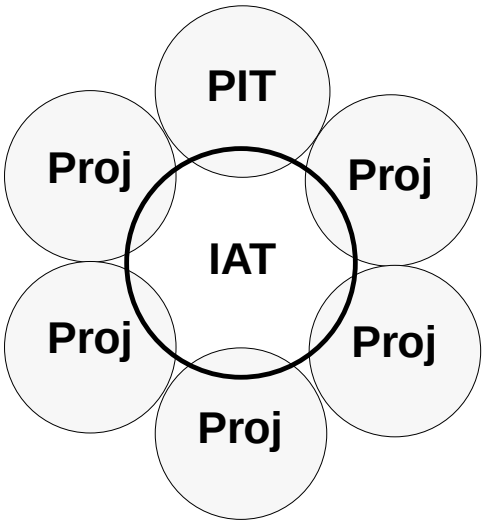
Resulting Context:

- Temporally recurring process “SWAT teams” which enlist practitioners in SPI
- The IAT focuses exclusively on the one improvement, leaving the rest of the PIT free to do other things while still “keeping tabs” on the IAT
- IAT members and their projects make good improvement pilot-test candidates
- Requires appropriate rewards/reinforcement to encourage participation and cooperation (and, or course, management support)

Known Uses:

- Raytheon *[Haley]*; Tinker Air Force Base *[Herbsleb,Carleton]*; SPI case studies *[Fowler,Rifkin]*

34. Improvement Action Teams *(cont.)*



35. Improvement follows Process

Context:

- Process follows Practice has been applied
- PIT or IAT is ready to start designing/implementing/deploying process changes

Problem:

What process should be used for improving the process itself?

Forces:

- Ideally, the process should be capable of encompassing self-improvement
- If it were this far along, many such improvements wouldn't be necessary
- Using policies and procedures different from those you have recommended damages your credibility within the development community
- It also indicates the process' inability to handle the existing range of projects
- But many SPI activities/concerns are very different from product development

36. Improvement follows Process *(cont.)*

Solution:

- *When plausible*, use the same process you're imposing/have already imposed
- New improvement proposals should take into account how they might be practiced for process development as well as product development
- Some things may make sense for products, but not the process (they may be different, or missing/extra between the two):
  - Look for common elements & abstract them into general policies/guidelines
  - Individual projects (including SPI) tailor these to their needs (within policy)

Resulting Context:

- Congruence between the words of the PITs and IATs with their own actions, and with the desired actions of the rest of the development community
- "Practicing what you preach" lends credibility to your efforts
- The process becomes adaptable enough for product *and* process development

Known Uses:

- Microsoft *[McCarthy], [Cusumano,Selby]*; SPI case studies *[Curtis], [Fowler,Rifkin], [Donaldsen,Siegel]*

37. Improvement follows Spiral

Context:

- Need an overall battle-plan to structure activities for incremental improvements
- This may apply to general SPI efforts by the PIT or specific efforts by an IAT

Problem:

What framework should be used to structure the varied activities of planning, implementation, assessment, and deployment for SPI?

Forces:

- Group-wide SPI efforts must be carefully planned if they are to succeed
- Many risks must be identified, evaluated, and appropriately addressed
- Omitting an important step or overlooking a key risk can result in project failure
- Too much planning & analysis can slow/impede progress (analysis paralysis)
- Too much action and not enough assessment may result in sloppy and ineffective efforts that eventually fail
- Even if a suitable balance of action and reflection is found, their order and frequency can make or break an SPI initiative

38. Improvement follows Spiral *(cont.)*

Solution:

- Impose a spiral model upon the process improvement lifecycle
- Use a variant of the Shewhart cycle of *Plan-Do-Check-Act* (espoused by Deming and in TQM circles)

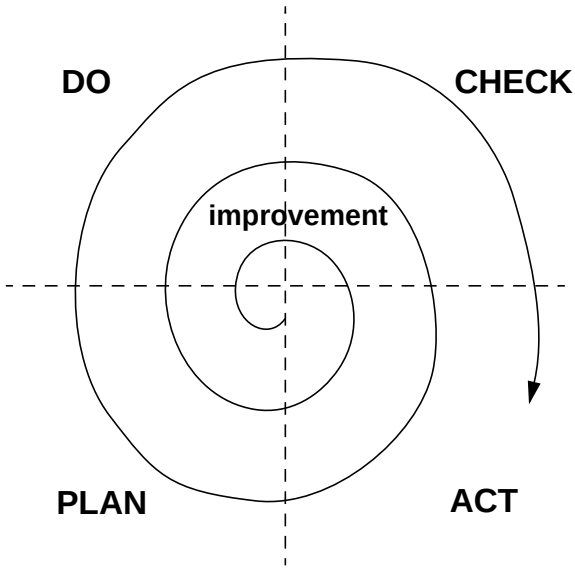
Resulting Context:

- A spiral framework for iteratively incorporating planning, assessment, and risk management activities into SPI
- The spiral model is used in a manner similar to that recommend for software
- The Shewhart cycle tailors the spiral model for use with SPI efforts (*[Grady]*)

Known Uses:

- Hewlett-Packard, *Plan-Do-Check-Act [Grady]*
- Kodak, *Plan-Do-Assess-Verify [Wiegers]*
- NORAD, *Analyze-Plan-Do-Check-Act [Wakulczyk]*
- SPI experience reports *[Kellner], [SPC]*; and reports of "real world" extension to SEI's IDEAL model: *Initiate-Diagnose-Enact-Assess-Leverage*, *[Jones,Kasunic] and [Radice]*

39. Improvement follows Spiral *(cont.)*



40. SPI Conclusions

- Process change entails cultural change!
- Process Improvement and Product Development are very similar yet very different, both of which are vitally important!

The *process patterns* shown here extol the similarities:

- A process *is* a product!
- The existing process is a legacy system
- SPI is a legacy systems-reengineering project
- Plan & Manage SPI projects much like development projects
- SPI procedures should closely resemble product development procedures
- Evolutionary/Incremental development seems to meet with greater success
- *Engaging customers early and often* in dialogues which regularly communicate status and feedback is a crucial element of success (and its absence is often a leading cause of project failures)

41. SPI Conclusions (cont.)

The *organization and communication patterns* shown here focus on the important social and cultural differences:

- Social organization and communication strategies for SPI must accommodate the fact that *the customer actually lives under the same roof* as the enterprise itself, co-habiting with all of its members
- Customer *communication & interaction issues are profoundly amplified* in SPI projects because the customers are members of the same cultural community as SPI project managers and architect
- As a result, *the organization's internal ecosystem is more sensitive* to the impact of improvement efforts since they effect changes in that very same culture

42. Open Issues

Still need answers to other important SPI questions:

- How do you successfully obtain senior management “buy in”?
- How do you convince others of both the real and perceived need for SPI?
- How do you create a shared mental model of the desired end-result?
- How should you setup rewards, incentives, and reinforcement?
- How should you solicit practitioner enrollment?
- How should you establish process ownership?
- How should you conduct training and education?
- How should you assess/evaluate SPI progress?
- What needs to be done differently for calendar-driven, architecture-driven, or documentation-driven organizations?
- What about groups in constant crisis or crisis-aversion mode?

We need patterns for all these and more before we have a comprehensive solution for initiating and sustaining SPI!

43. Why Patterns?

Software Patterns help us because they:

- Solve “real world” problems
- Capture domain expertise
- Document design decisions and rationale
- Reuse wisdom and experience of master practitioners
- Convey expert insight to novices
- Form a shared vocabulary for problem-solving discussion
- Show *more* than just the solution:
  - context (when and where)
  - forces (trade-off alternatives, misfits, goals+constraints)
  - resolution (how and why the solution balances the forces)

44. Summary - What Patterns Are *Not*

Software Patterns are ***not*** ...

- Restricted to software design or object-oriented design
- Untested ideas/theories or new inventions
- Solutions that have worked only once
- Any old thing written-up in pattern format
- *Abstract* principles or heuristics
- Universally applicable for all contexts
- A “silver bullet” or panacea

45. Summary - What Patterns *Are*

Software Patterns ***are*** ...

- *Recurring* solutions to common problems of design
- *Practical/concrete* solutions to real world problems
- *Context* specific
- “*Best-fits*” for the given set of concerns/trade-offs
- *“Old hat”* to seasoned professionals and domain experts
- A *literary form* for documenting best practices
- A *shared vocabulary* for problem-solving discussions
- An effective means of (re)using, sharing, and building upon *existing wisdom/experience/expertise*
- *Massively overhyped!*

46. SPI Books & Publications

- **Creating a Software Engineering Culture**
  Karl E. Wiegers, Dorset House, 1996
  (see also *http://www.frontiernet.net/~kwiegers/*)
- **Successful Software Process Improvement**
  Robert B. Grady, Prentice-Hall, 1997
- **Cultivating Successful Software Development: A Practitioner's View**
  Scott E. Donaldsen, Stanley G. Siegel, Prentice-Hall PTR, 1997
- **Software Engineering Process Group Guide**
  Priscilla Fowler, Stan Rifkin, Carnegie Mellon University
  SEI Technical Report CMU/SEI-90-TR-024, September 1990
  available online from *http://www.sei.cmu.edu/products/publications/doc.list/index.html*
  (see also *http://www.sei.cmu.edu/technology/cmm/cmm.articles.html*)
- **Improving the Software Process Through Process Definition and Modeling**
  Software Productivity Consortium, International Thomson Computer Press, 1996
- **SEPG Conference Proceedings**

47. Books on Organizational/Culture Change

- **Managing at the Speed of Change**, Daryl Conner, Villard Books, 1993
- **Leading Change**, John Kotter, Harvard Business School Press, 1996
- **Quality Software Management Volume 4: Anticipating Change**
  Gerald Weinberg, Dorset House, 1997
- **Changing the Way We Change**, Jeanenne LaMarsh, Addison-Wesley, 1995
- **Beyond the Wall of Resistance**, Rick Maurer, Bard Press, 1996
- **Battling the Barriers to Success**, Joan Klubnik and Marlene Roschelle, Irwin, 1996
- **Agents of Change**, Barbara M. Bouldin, Yourdon Press, 1989
- **Corporate Lifecycles**, Ichak Adizes, Prentice-Hall, 1988
- **Reengineering the Corporation**, Michael Hammer and James Champy, Harper, 1993
- **Beyond Reengineering**, Michael Hammer, Harper, 1996
- **The Fifth Discipline**, Peter M. Senge, Currency-Doubleday, 1990
- **The Fifth Discipline Fieldbook**, Senge et. al., Currency-Doubleday, 1994

See Amazon books for more than a hundred other references on the subject at:
http://www.amazon.com/exec/obidos/Subject=Organizational%20change/4907-6944902-566134

48. Pattern Resources - Books

- **A Pattern Language: Towns, Buildings, Construction** (APL)
  Christopher Alexander; Oxford University Press, 1977
- **The Timeless Way of Building** (TTWoB)
  Christopher Alexander; Oxford University Press, 1979
- **Design Patterns: Elements of Reusable Object-Oriented Software** (GoF)
  Gamma, Helm, Johnson, Vlissides; Addison-Wesley, 1994
- **Pattern-Oriented Software Architecture: A System of Patterns** (POSA)
  Buschmann, Meunier, Rohnert, Sommerlad, Stal; Wiley and Sons, 1996
- **Pattern Languages of Program Design** (PLoPD1)
  Coplien and Schmidt (editors); Addison-Wesley, 1995
- **Patterns of Software: Tales from the Software Community**
  Richard Gabriel; Oxford University Press, 1996
- **Analysis Patterns: Reusable Object Models**
  Martin Fowler; Addison-Wesley, 1996
- **Pattern Languages of Program Design 2 (PLoPD2)**
  Vlissides, Coplien, and Kerth (editors); Addison-Wesley, 1996

49. Pattern Resources - Online

- Patterns Home Page, *http://www.hillside.net/patterns/*
- Patterns Discussion FAQ, *http://g.oswego.edu/dl/pd-FAQ/pd-FAQ.html*
- Ward Cunningham's WikiWikiWeb, *http://c2.com/cgi/wiki?WelcomeVisitors*
- Portland Pattern Repository, *http://www.c2.com/pp/*
- AGCS Patterns Page, *http://www.agcs.com/patterns/*
- Jim Coplien's OrganizationPatterns Front Page (a WikiWikiWeb clone), *http://www.www.bell-labs.com/cgi-user/OrgPatterns/OrgPatterns*
- Patterns Mailing Lists, *http://www.hillside.net/patterns/Lists.html*
- Cetus Links: Patterns, *http://www.objenv.com/cetus/oo_patterns.html*
- Brad's Pattern Links: *http://www.enteract.com/~bradapp/links/sw-pats.html*
- Brad's Patterns Intro: *http://www.enteract.com/~bradapp/docs/patterns-intro.html*
- Luke Hohmann's Patterns Intro: *http://members.aol.com/lhohmann/papers.htm*
- Doug Lea's OOD Patterns Intro: *http://gee.cs.oswego.edu/dl/ca/ca/ca.html*

50. The Chicago Patterns Group (TCPG)

- Meets the 1st and 3rd Tuesday of every month
- Informal gathering from 7pm-9pm at Borders Books
  - at the northeast corner of Golf (IL-58) and Meacham roads
  - 1540 Golf Road, Schaumburg, IL 60173, (847)330-0031
- Read & Discuss all kinds of Software Patterns
  - Patterns of Software Design, Analysis, Process, Organization, etc.
- Have been meeting (semi-monthly) since January 1997
- Newcomers are always welcome!
  - Email *<bradapp@enteract.com>* to be added to the email notification list for TCPG meetings and special events
- See the TCPG home page for info and status
  - *http://c2.com/cgi/wiki?ChicagoPatternsGroup*

51. Presenter Information

Professional

Brad Appleton, Senior Software Engineer
Motorola Automotive and Industrial Electronics Group
4000 Commercial Avenue, Northbrook, IL 60062
email: *Brad_Appleton-GBDA001@email.mot.com*

Personal

Brad Appleton, Software Tools Developer
email: *bradapp@enteract.com*
web: *http://www.enteract.com/~bradapp/*
(3500+ www links to software engineering & computer science at the above URL)

Papers available at the *"Documents"* section of my webpage:

- *Patterns for Conducting Process Improvement*
- *Patterns in a Nutshell: The "bare essentials" of Software Patterns*
- *Patterns and Software: Essential Concepts and Terminology*