Experience Factory and Organizational Learning

Professor Jyrki Kontio

Email: jyrki.kontio@tkk.fi
http://www.sbl.tkk.fi/jkontio/

SOFTWARE BUSINESS LABORATORY
Helsinki University of Technology
http://www.sbl.tkk.fi/

HELSINKI UNIVERSITY OF TECHNOLOGY

My Background with Vic

- Worked at Nokia in knowledge-based systems since 1986
- Head of the software engineering research group at Nokia Research Center – but with insufficient knowledge and background
- Plan for a Ph.D. ➔ what is the best group in the world? ➔ Bill Curtis: ”Contact Vic”
- Research at UMD 1994 – 96:
  - Risk management
  - COTS evaluation
  - Process management, Experience Factory
  - SEL support: DB redesign, etc.
- Dissertation completed shortly after leaving Maryland .. ;-)
- Currently heading the Software Business Laboratory at Helsinki University of Technology

Main Points

- Experience Factory as a scientific paradigm
- Industrial view on the use of the Experience Factory

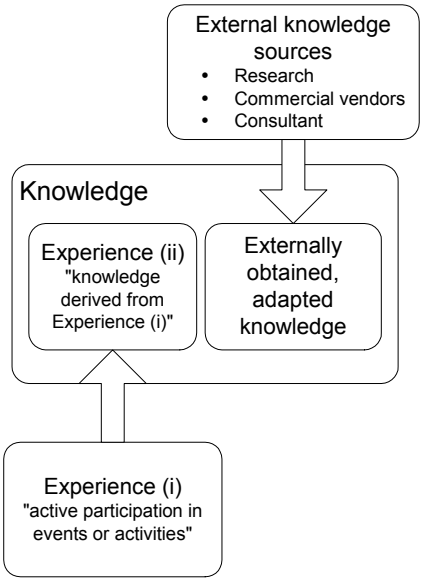
- Outline
  - Experience vs. Knowledge
  - Primary Functions of the Experience Factory
  - Experience Factory as a Paradigm
  - Experience Factory and Maturity Models
  - Impact of Experience Factory
  - Biggest Challenges
  - Personal Lessons Learned from Vic

Experience vs. Knowledge

Experience can be defined as

(i) "active participation in events or activities, leading to the accumulation of knowledge or skill: a lesson taught by experience; and

(ii) "knowledge or skill so derived"



Source: Kontio, 2001

Experience vs. Knowledge: Revised Definitions

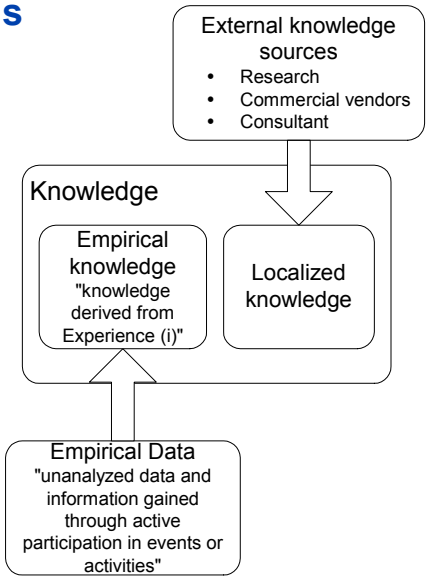
Experience can be defined as

(i) "active participation in events or activities, leading to the accumulation of knowledge or skill: a lesson taught by experience; and

(ii) "knowledge or skill so derived"

Knowledge is defined as

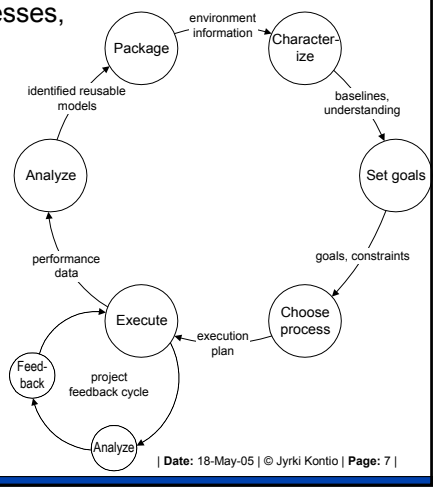
- "familiarity, awareness, or understanding gained through experience, study, or analysis"



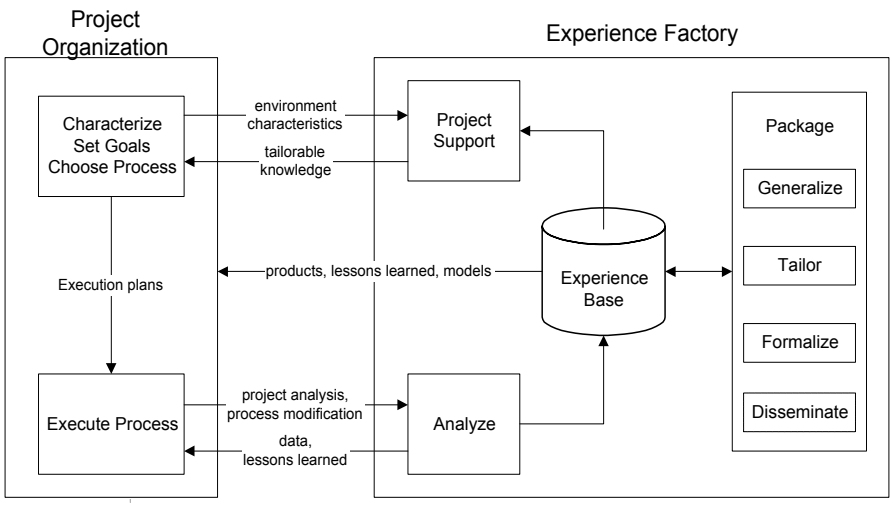
Source: Kontio, 2001

The Quality Improvement Paradigm Cycle

1. **Characterize**. Understand the environment based on the available data, models, experience, and insights.
2. **Set goals**. Set quantifiable goals for the project and organizational performance and improvement.
3. **Choose process**. Choose the processes, tools and techniques appropriate for the project.
4. **Execute**. Perform the process.
5. **Analyze**. Analyze the data and the information to evaluate current practices, determine problems, record findings, and make recommendations for future projects.
6. **Package**. Consolidate the experience gained in the form of new or updated models, documents and other forms of knowledge and store this knowledge in the experience base.



Experience Factory



Experience Factory Principles

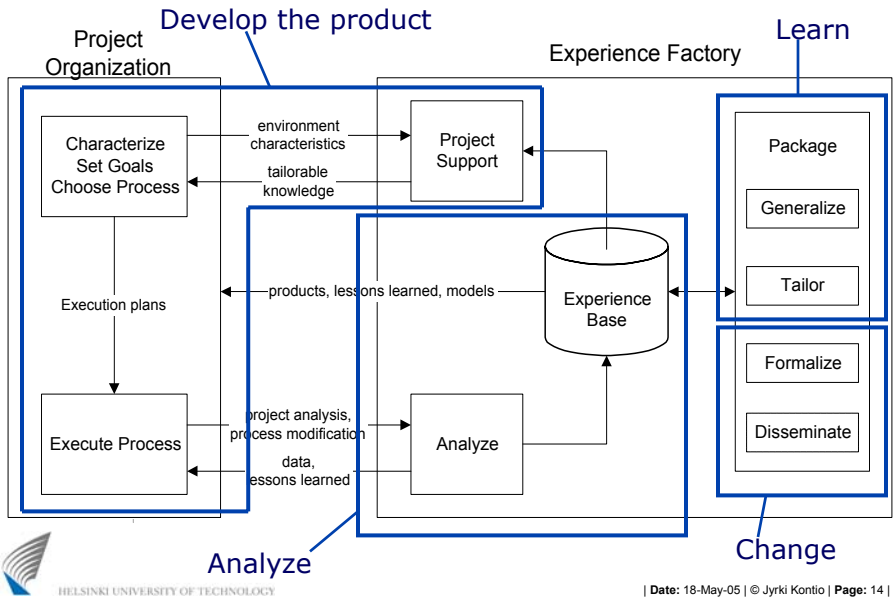
- Separation of responsibilities between product development and improvement;
- Systematic capture and accumulation of knowledge;
- Continuous learning from experience through measurement, data collection, analysis and synthesis; and
- Systematic reuse of knowledge through packaging and dissemination.

Primary Functions of the Experience Factory

1 Develop Software Product
- 1.1 Plan software development project
- 1.2 Manage and control software development project
- 1.3 Develop, test and deliver software product

2 Process Improvement
- 2.1 Understand
  - 2.1.1 Model and document
  - 2.1.2 Measure
- 2.2 Learn
  - 2.2.1 Analyze current characteristics
  - 2.2.2 Acquire knowledge
    - 2.2.2.1 Empirical studies
    - 2.2.2.2 Outsource knowledge
  - 2.2.3 Synthesize knowledge
  - 2.2.4 Validate knowledge
- 2.3 Change
  - 2.3.1 Plan change
  - 2.3.2 Implement change
    - 2.3.2.1 Package
    - 2.3.2.2 Disseminate
    - 2.3.2.3 Institutionalize

EF and the Primary Functions



Experience Base

- Process models
- Experience reports
- Research issues
- Prediction models
- Enactment data
- Assessment & benchmarking results

➔Process models embody current knowledge and best practices

➔Experience reports contain data and analysis from empirical studies

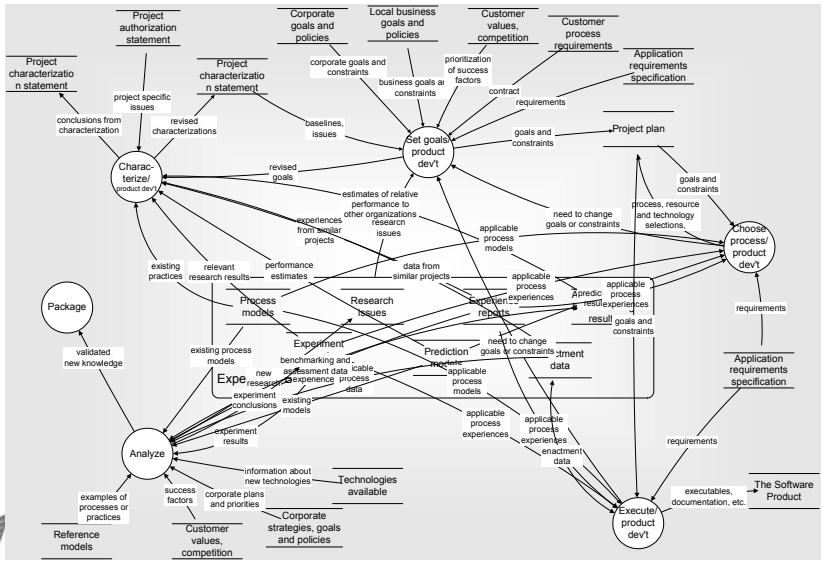
➔New questions and need for research

➔Formalized models and tools to support planning

➔Data on past projects

➔Results of assessments and benchmarking

What Really Goes On in the Experience Factory?



What is a "paradigm"?

- The contemporary meaning by Thomas Kuhn (1962):
  - "set of accepted principles of scientific practice that are common to a group of researchers", such as laws, theory, experimental design, and instrumentation.
  - Two main characteristics of a paradigm are
    (i) paradigm represents achievements that are sufficiently unprecedented to attract scientists from other paradigms and
    (ii) the paradigm is sufficiently open-ended to have problems to be resolved.
- Our definition:
  "***a point of view in which some unique assumptions, theories, methods, guidelines and research methods are stated uniformly***" (Kontio, 1995)

QIP and EF as a Paradigm

- Defined with Vic Basili's research group*
  - Assumptions
  - Methods
  - Research methods
- Based on a review of papers on QIP and EF

* Acknowledging insights from Vic Basili, Carolyn Seaman, Walcelio Melo, Gianluigi Caldiera and Giovanni Cantone, among others

Assumptions

1. Continuous learning is essential for all evolutionary fields, such as software engineering.
2. Continuous, sustained improvement is not possible without understanding of the current situation and environment.
3. Measurement and modeling are essential for understanding and learning.
4. All knowledge is potentially reusable and, therefore, should be explicitly represented.
5. Improvement and organizational goals must be explicitly stated and measured.
6. All software development knowledge must be localized. We do not yet have universal models for software quality or productivity but if and when such universal models are identified, they will need to be localized as well.

Assumptions (continued)

6.1. Knowledge is reusable within the same domain it was initially formulated. If it is reused in other domains or situations, the success of this reuse is strongly dependent on the understanding of the similarities and differences between the situations.

6.2. An organization must build up its own understanding of its products and processes, based on measurement, modeling and analysis.

6.3. The measurement and modeling objectives vary and actual metrics and models are dependent on these objectives.

6.4. Improvement objectives are specific to each organization and, among other things, depend on business goals and strategies, competitive situation, organization's current strengths and weaknesses, customer needs and preferences, and the technologies available.

6.5. The type and characteristics of the software process depend on the organizational and improvement objectives.

Methods

7. The QIP method represents an effective method for creating localized knowledge for software development.
8. The Experience Factory represents an effective model for implementing a quality improvement system that aims at creating localized knowledge.
9. The Experience Base, as a part of the Experience Factory, represents an effective way to document and accumulate localized knowledge.
10. The GQM method represents an effective method for defining metrics that are goal and situation dependent.

Research Methods

11. Software engineering research must be empirical, i.e., theories must be validated by observations, experiments, surveys and data collection.
12. Good experimental design improves the confidence and usability of results from experiments.
13. Both qualitative and quantitative techniques will need to be used in software engineering research.

What is the Use of ”EF as a Paradigm”?

- Research is done by a community of researchers
  - The community needs to know what it is and where it is going
- People in other communities have a better chance at understanding our ”paradigm”
  - We can ”spread the word” to other researchers
- The community can debate, refine and improve its underlying characteristics in order to improve

Experience Factory and Maturity Models

Experience Factory	Maturity Models
All knowledge is local and needs to adapted	There are some best practices that are applicable to all organizations
Primarily source of new knowledge: systematic learning through empirical studies	Best practices are well documented and can be obtained and transferred to SW organizations
Organization-specific objectives determine what improvements are most effective	There is a pre-defined order in which improvements need to be made

➔ Both approaches are needed, Maturity Models for speed and impact, EF for sustainable competitive advantage

Impact of the Experience Factory

- EF and QIP have a had a major impact in the academic community
  - The importance of empirical studies have been widely accepted
  - The rigor in empirical work has increased substantially
- Several leading organizations have adapted and used EF and QIP
  - NASA, Nokia, Motorola, DaimlerChrysler, Boeing, ABB, ...
- The empirical approach has improved the validity of the research in the whole software engineering field

Characteristics of Industry vs. Academia

Academia

- The established paradigm rules
- Slow review cycles
- Search for ”truth”
- Strong peer review, but sometimes lack of empirical rigor

Industry

- Identify business benefit clearly
- Show fast results
- Good enough solution is sufficient
- Things change quickly
- Little time for deep analysis
- Who you know is important
- Commitment for long-term work when crucial for business

Implications

Academia	Industry
▪ Established paradigm rules ▪ Slow review cycles ▪ Search for "truth" ▪ Strong peer review, but sometimes lack of empirical rigor	▪ Identify business benefit clearly ▪ Show fast results ▪ Good enough solution is sufficient ▪ Things change quickly ▪ Little time for deep analysis ▪ Who you know is important ▪ Commitment for long-term work when crucial for business

➔Build you case on evidence

➔Be specific to your situation, then generalize

➔Focus on results and get them quickly

Challenges in Leveraging the Experience Factory

- It is known and used, but not widely used in industry
- Why?
  - Systematic learning is hard
  - Companies want quick, cheap, and easy solutions
  - People do not know about EF
  - We have not packaged the EF well enough
  - Most organizations are in such bad shape that anything will be an improvement

Biggest Challenges for the ”EF Community”

- Laboratory vs. practice
  - Empirical work often involves limitations and constraints, can results be transferred to practice?
- Speed of learning vs. speed of change
  - Business needs results quickly, experimentation takes time
  - Technology and context change quickly, will ”old knowledge” become obsolete?
- Overhead of learning and measurement
  - ”Experience Factory costs 10-12% of R&D”
  - Is this too much for some companies?
- What if decisions based on intuition (and not on sound, empirical evidence) is good enough in business?
  - Managerial reactions might compensate for initially wrong decisions

Call for Even Greater Impact

- We need to communicate, package and deploy the Experience Factory even better
- We need to make basic training on empirical methods and Experience Factory more easily accessible to practitioners
- We need to conduct our research faster and aim at even bigger impact – while not making any compromises on the quality of the research
- Make a contribution to the community, it will have good ”ROI” for you and for the community

Personal Lessons Learned from Vic

Do the right thing

Find focus in your work

Start from goals

Have fun

Vic,
thanks for making
an impact!

References

- V. R. Basili, Software Development: A Paradigm for the Future (keynote address), 1989. Proceedings of the 13th Annual Computer Software and Applications Conference (COMPSAC).
- V. R. Basili and H. D. Rombach, The TAME Project: Towards Improvement-Oriented Software Environments, IEEE Transactions on Software Engineering, vol. 14, 6. pp. 753-778, 1988.
- V. R. Basili and H. D. Rombach, Support for comprehensive reuse, Software Engineering Journal, vol. September. pp. 303-316, 1991.
- T. S. Kuhn. The Structure of Scientific Revolutions, Chicago: University of Chicago Press, 1970.
- Kontio, J., A Process Engineering Framework, in: Advances in Computers, M.V.Zelkowitz, ed. Academic Press, pages 36-108, 1998.
- Kontio, J., Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation. Doctoral dissertation. (2001), Helsinki University of Technology, publisher: Center of Excellence, ISBN: 952-5136-22-1.