Software Process Optimization (Part-2)

Masao ITO

Nil Software
VCAD Solutions

tomorrow’s seed (1993)
Kishida-san
Contents

• General problem solving framework and software engineering technique

• Steps of software process optimization
  • Monitoring and evaluation of SPO’s work and feedback into the organization

• Case study
1. Motivation
Motivation

Plan

Original Goal

++

Real

Final Goal
Motivation

• Our SPO approach is technology-oriented. It is based upon software engineering techniques. We are software engineers and able to apply our technical knowledge to solve various process problems.

• During the long history of software engineering various useful technologies have been developed. So, it is natural to apply way those technologies to solve process problems we encounter in the course of software development projects.
Transformation into an immaterial labor
2. Outline of SPO
Characteristics of SPO

- Participatory approach
- Using techniques of software engineering
- Comprehensive approach
  - whole and part
  - analysis and creation (i.e. solving)
- Enjoyable!
Characteristics of SPO

What is the difference between SPI and SPO

- SPI based on SPA is good at finding the deficiency comparing with the average software process, if it exists.

- **SPO** is systemized, it helps many SEPG people to find the better way for changing process. It covers whole software development process as human activities differing among projects and organizations.
SPO FRAMEWORK
relating topics

KAOS
CRC card
UML

Software
Engineering

Process
Techniques

Organization
Management

Self Communion

15508
PCM

Activity
Theory

Base Model
KAOS

- KAOS (Knowledge Acquisition in autOmated Specification) is a method for modeling requirements

KAOS

goal, interference, expectation

- Goal
- Develop Spec.
- Get Customer Request
- Define Spec.
- Feasibility Problem
- Expectation
- Customer Understanding
- Interference
- Misunderstanding
CRC card

- CRC (Class Responsibility Collaboration) is a method for retrieving objects in bottom-up manner.
CRC card transformation
UML

- UML is a modeling language for the software development
Activity Theory

• “Activity Theory is a philosophical and cross-disciplinary framework for studying different forms of human practices as developmental processes, with both individual and social levels interlinked at the same time.” Kuutti (1996)

• The way to articulate contradiction
Activity Theory
Engeström’s activity systems thinking

- Subject
- Object
- Outcome
- Rules
- Community
- Division of labor

[Diagram showing relationships between subjects, objects, outcomes, rules, community, and division of labor with mediating artifacts at the center.]

schema
Activity Theory
5 principles

• Collective, artifact-mediated and object-oriented nature
• Multi-voicedness
• Historicity
• Contradiction as sources of change and development
• Expansive transformation
Activity Theory
interaction of activities
3. Details of Software Process Optimization
Steps

1. Participatory Planning
2. Mission Definition
3. Analysis/Resolving Support
   - Closed-World Model
   - Open-World Model
4. Monitoring/Evaluation
5. Resolving

Library
Participatory Planning
Participatory Planning

outline

• Stakeholders analysis
• Abstract problems analysis
• Objectives analysis
• Summarize in SPO Activity Matrix (SAM)
Participatory Planning
analysis of stakeholder

- Confirm the expected goal
- List on cards all individuals, organizations within the expected goal
- Sort the cards by category
- Select important stakeholders
- Analyze the selected stakeholders in detail
- Tentatively select a target group
Participatory Planning

stakeholder on the card

<table>
<thead>
<tr>
<th>Basic information</th>
<th>Objective information about socio-cultural structure, technical ability, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem/Weaknesses</td>
<td>What kinds of problems and weakness are there?</td>
</tr>
<tr>
<td>Needs</td>
<td>What are there needs?</td>
</tr>
<tr>
<td>(Potential) strengths</td>
<td>What advantageous quality or resources already exist? What is the potential capabilities?</td>
</tr>
<tr>
<td>Action to take</td>
<td>What kinds of actions can be taken in the above conditions?</td>
</tr>
</tbody>
</table>
Participatory Planning
abstract problem analysis

- Identify a core problem
- Write the problems that are direct causes of the core problem and place them under the core problem
- Repeat above procedure
- Write the direct effects of the core problems above the core problem
- Repeat above procedure
Participatory Planning
abstract problem analysis

- Rework occurred in design phase
- Rework occurred in implementation phase
- Rework occurred in test phase

Specifications are often misunderstood

- Spec. has ambiguity
- It's hard to know the right place of spec.
- Old spec. was referred
- Future version was referred

Analyst has their own style of writing

direct effect

core problem

direct causes
Participatory Planning
objectives analysis

- Identify a core objective
- Write the potential direct means for the core objective and place them under the core objective
- Repeat above procedure
- Write the direct end of the core objective above the core objective
- Repeat above procedure
Participatory Planning

objectives analysis

- No rework will occur in design phase
- Specifications are understood rightly
- Format of spec is defined
- Analysts are educated as for writing spec

- No rework will occur in implementation phase
- List of spec to be referred is maintained
- Designer have to write CM no. in doc.

- No rework will occur in test phase
- core objective
- direct ends
- direct means
Mission Definition
No rework will occur in design phase

Specifications are understood rightly

Format of spec is defined

List of spec to be referred is maintained

Analysts are educated as for writing spec

Designer have to write CM no. in doc.

No rework will occur in implementation phase

No rework will occur in test phase

overall goal

purpose

outputs
Mission Definition
SPO Activity Matrix (SAM)

- Overall Goal
  - What will be aimed at after the mission purpose is achieved?
- Purpose
  - What should the mission achieved within the mission duration?
- Outputs
  - How should the mission achieve the purpose?
- Actions
  - What should be done concretely to produce the outputs?
- Input
  - Personnel, materials, equipment, facilities, and funds
- Preconditions
  - Conditions that must be fulfilled before the mission gets underway.
## Mission Definition

### SPO Activity Matrix (SAM)

<table>
<thead>
<tr>
<th>Summary</th>
<th>Verifiable Indicator</th>
<th>Means of Verification</th>
<th>Important Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td></td>
<td><strong>Inputs</strong></td>
<td>Pre-conditions</td>
</tr>
</tbody>
</table>

**Mission Name:**

**Target Area:**

**Duration:**

**Target Group:**
Mission Definition

SAMs

- SAM might be revised by rethinking of plan or change of situation (it would occur constantly)
Analysis/Resolving Support

Closed-world Model
Open-world Model
Resolving
Closed-World Model
Closed-World Model

- Dig down the activity deeper within our current scope
Closed-World Model steps

- Define the adopted activity
- Write Object-Attribute-Action (OAA) statements
- Describe Quantitative Change graph (QC graph)
Closed-World Model
Object-Attribute-Action

Developer

Analyst

Share Spec.

Software Spec.
{id, text}

Functional Spec

Non-Functional

code

1..n

1..2

refer

create

1..n

next

prev
In QC-graph, you do thought experiment quantitatively.

Off course, this data is sample.

QC: Qualitative Change
Open-World Model
Open-World Model

- Sketch the problem situation and the ideal solution
- Apply particles to bridge the gap between problem and ideal solution
- Describe and/or tree
Open-World Model
Real State

Developer A

Developer B

Analyst
Open-World Model
ideal state

Developer A

Developer B

Analyst
Open-World Model
solution exists between real and ideal state

Developer A

Developer B

Analyst
Open-World Model

AND/OR tree

- Particles can deliver the right item as a spec to the designer
  - Particles know which version is needed by the designer
  - Particles know about the version that analyst want to apply
  - Particles hold the every item of the specification
    - Item has the version information
    - Analyst tells the version when passing the item
Resolving
Resolving

- Analyze Uniqueness
- Examine solutions by dimensionality/pluralization/distribution/transduction
- Generate concepts
- Consolidate concepts and find out ONE solution
Resolving characteristics

- Action
- Relation and number of objects
- Role of each object
- Attribute of each object
Closed-World Model
Object-Attribute-Action
Resolving

uniqueness of space-time

• Space
  • The size of each characteristics in multidimensional space

• Time
  • Variation of characteristics moment by moment
Monitoring/Evaluation
Monitoring/Evaluation

- **Monitoring** is to check the progress of an active mission based on the SAM, and modify the plan if necessary.

- **Evaluation** is to check the accomplished mission by the five evaluation criteria.
Monitoring/Evaluation

aim of evaluation

- Get the lesson-learned or future direction from the mission, not for criticizing
## Monitoring/Evaluation

### Evaluation Table

<table>
<thead>
<tr>
<th>Overall Goal</th>
<th>Relevance</th>
<th>Effectiveness</th>
<th>Efficiency</th>
<th>Impact</th>
<th>Sustainability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Monitoring/Evaluation

#### Evaluation Table

<table>
<thead>
<tr>
<th>Summary</th>
<th>Verifiable Indicator</th>
<th>Means of Verification</th>
<th>Important Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>Inputs</td>
<td></td>
<td>Pre-conditions</td>
</tr>
</tbody>
</table>

#### SAM

**Evaluation Table**

<table>
<thead>
<tr>
<th></th>
<th>Relevance</th>
<th>Effectiveness</th>
<th>Efficiency</th>
<th>Impact</th>
<th>Sustainability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Monitoring/Evaluation

5 evaluation criteria

- Relevance
  - Have there been any changes in the organization policy?
- Effectiveness
  - Is the purpose expected to be achieved by the end of period?
- Efficacy
  - Are the “inputs” being utilized properly to produce “Outputs”?
- Impact
  - Is the mission producing any negative effect?
- Sustainability
  - Is the target group properly prepared to sustain the benefits of their process after the mission end?
Library

- A repository for re-use in other cycles. It includes the output of mission:
  - SAMs (SAM0, SAM1, ... SAMn)
  - Triangles of activity
  - OAA
  - Evaluation results
- It becomes the asset of your organization