

Process improvement using XDDP

- Application of XDDP to the Car Navigation System -

Eiji Nakai

DENSO TECHNO Co.,Ltd

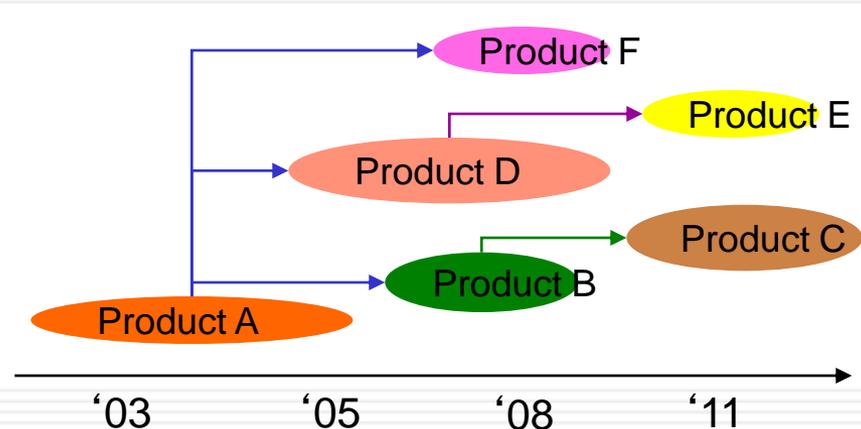
1. Introduction
2. Problems in Conventional Development Process
3. What is XDDP?
4. Case 1. Application to General Project
5. Case 2. Application to PWAT(*)
6. Conclusions

(*) PWAT : Project Without Accumulated Technical information

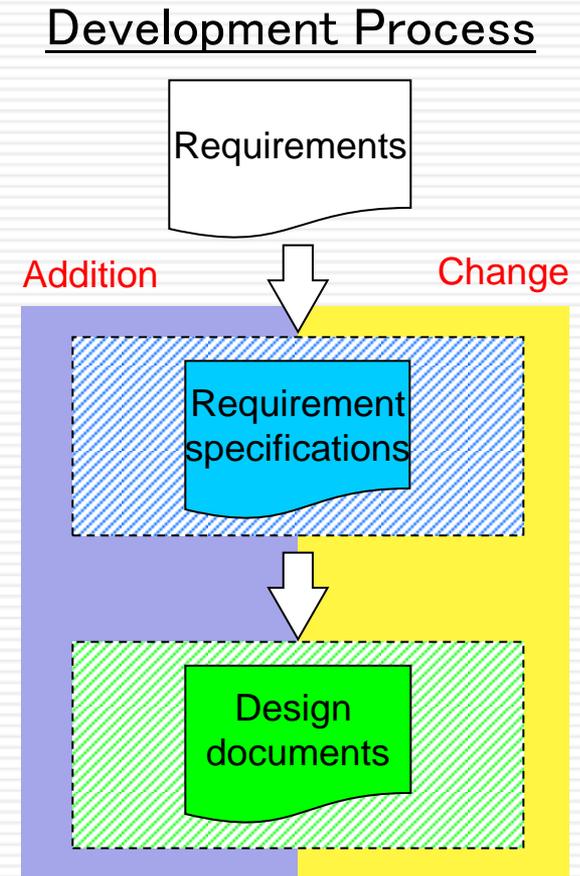
- Current situation in car navigation software development
 - Number of functions and size of software are increasing rapidly
 - Many software variations are needed
 - Higher quality is required
 - Development period is getting shorter

- Most of developments of navigation software are “enhancement - based development”
 - Develop new product based on existing product
 - Add new functions
 - Make improvements of existing functions

Product Deployment



- Use V-model
 - Development process to develop new software product
- “Addition” and “Change” are contained in enhancement - based development
 - “Addition”
 - Add new functions to base software
 - “Change”
 - Change existing functions in base software

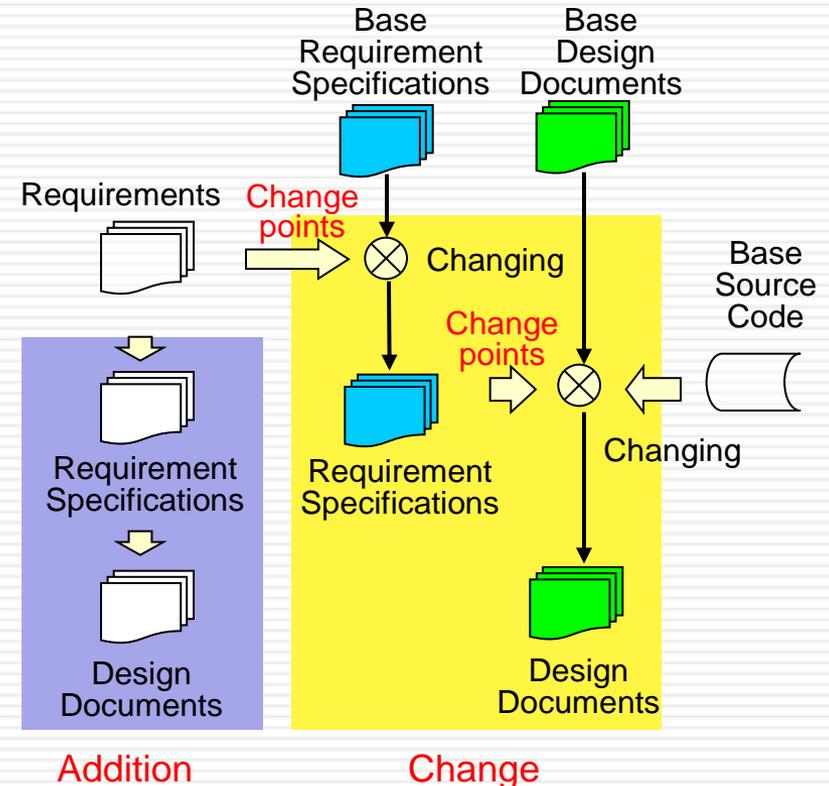


- It is difficult to identify all change points
 - One change point causes other changes
 - Adding a new function affects existing functions
- Source code becomes complicated by changes
 - Source code maintenance is not easy
- Design policy or change background is not always described clearly in the documents
 - Implementation of the change depends on engineers' knowledge

Problems are caused by “Change”

1. Change points are scattered in base documents
 - It is difficult to detect related changes
 - Review does not have much effect with such documents
2. Some engineers change the base source code without enough analysis
 - It leads to degrading other functions
 - The change points are not always appropriate

Details Conventional Process



Development Process is not suitable for “Change”

- PWAT (Project Without Accumulated Technical information)

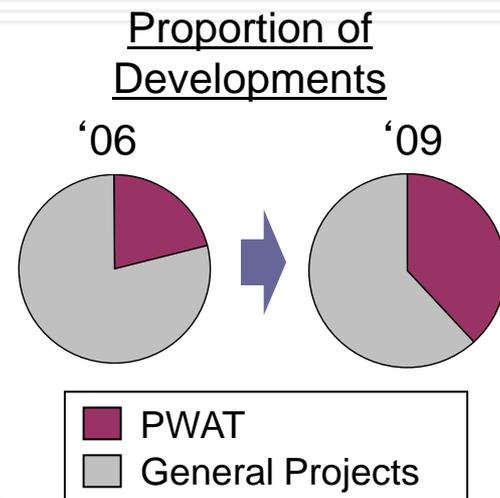
- No engineers have enough information about base software in the project

- Problems of PWAT

- Quality in investigation of base software depends on engineers' experience and intuitions

⇒ • Investigation is insufficient in coverage

- It takes too much time to analyze the base software
- Documents on the result of investigation are insufficient



Conventional Investigation does not meet PWAT

- XDDP is a software development process focused on Changes
- Advantages
 - Change information is arranged and described properly for the development
 - All related changes are detected through the development process

(1) Two independent processes

“Addition” and “Change”

(2) Specification techniques

USDM specification description

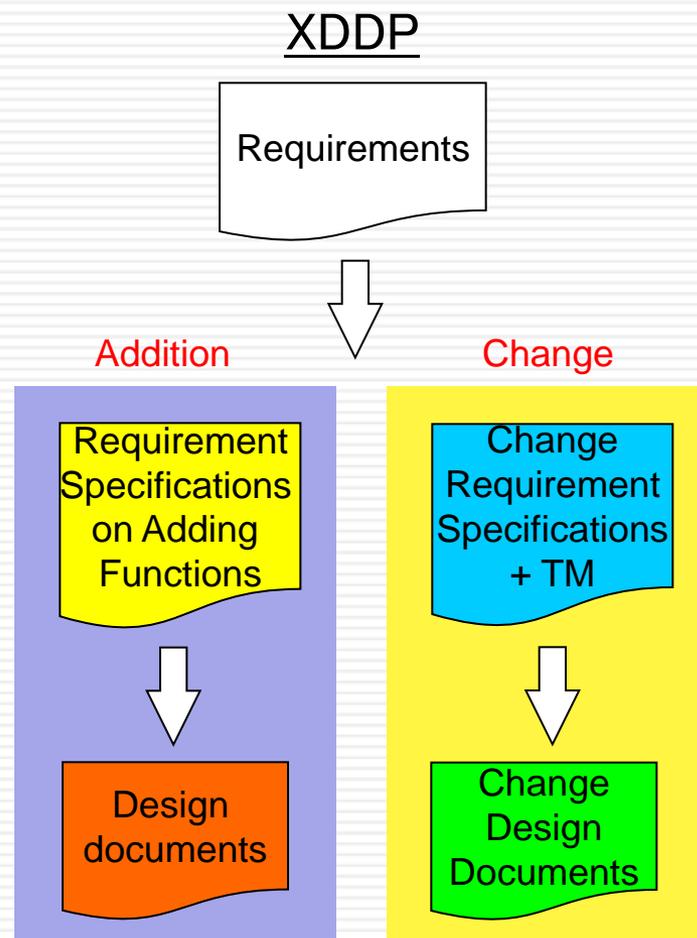
(*) Universal Specification Describing Manner

(3) Documents focused on change

“Change Requirement Specifications”

“Traceability Matrix (TM)”

“Change Design Documents”



(1) Two independent processes

- “Addition” and “Change”

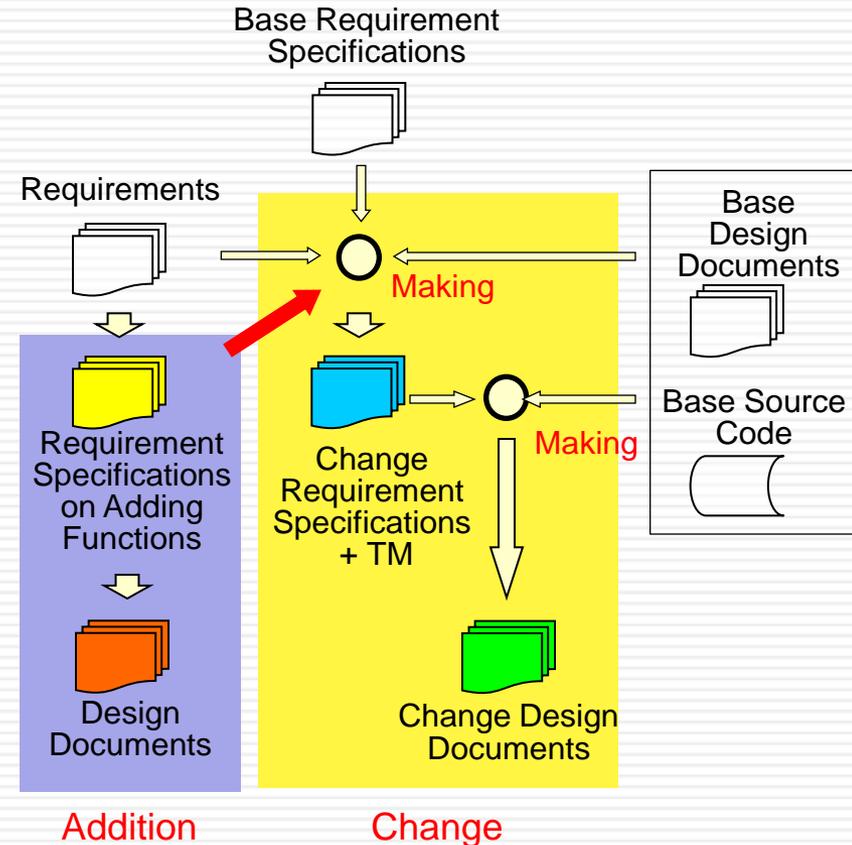
- Addition

- The process is the same as new development process

- Change

- Make new documents described about only changes
- Additional Function is treated as one change
- Change the source code after all change points are identified

Outline Flow of XDDP



Detect and describe all change points

• USDM Format

1. Describe requirements and specifications structurally

- “Requirement”
series of behaviors
- “Specification”
concrete behaviors to realize requirement

2. Clarify the reason

- The reason is necessary to detect proper change points

3. Describe change points by

“before/after” in “Change Requirement Specifications”

Requirement	Req.1	Requirement		
	Reason	Backgrounds or Objecti	3	
	Comment	-----		
1	Branch Requirement	Req.1-1	Branch Rquirement	
		Reason	2	
		Comment		
	Specifcaiton	<Group A>		
		Req.1-1-1	Specification	
		Req.1-1-2		
		<Group B>		
		Req.1-1-3		
		Req.1-1-4		
		Branch Requirement	Req.1-2	Branch Rquirement
	Reason			
	Comment		-----	
	Specifcaiton	<Group C>		
		Req.1-2-1	Specification	
Req.1-2-2				

USDM prevents missing change points

- Describe all change points in three types of documents before changing source code

[1] Change Requirement Specifications

- What and Why should we change ?

[2] Traceability Matrix (TM)

- Where should we change ?

[3] Change Design Documents

- How should we change ?

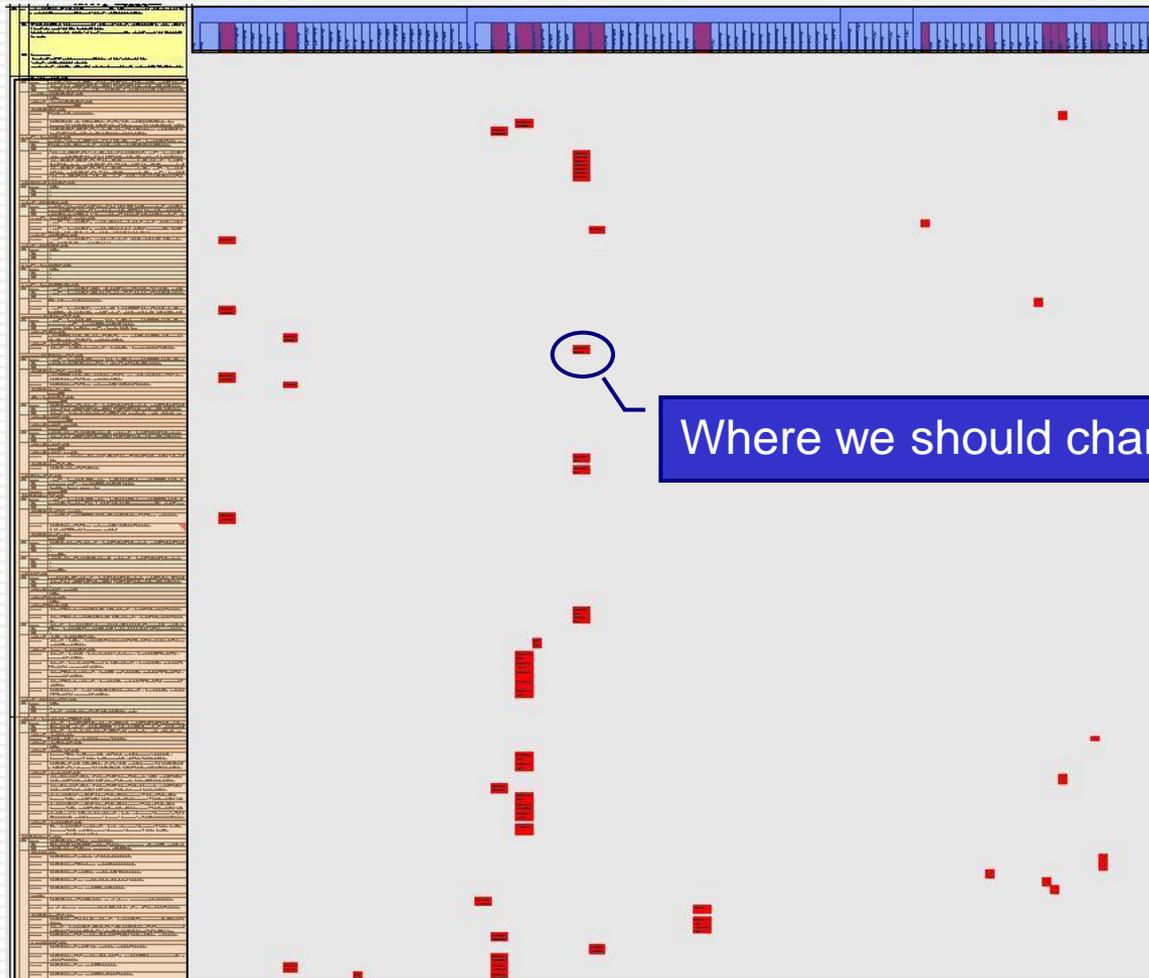
Change Requirement Specifications

TM

Requirement			Module A	SourceA.c	SourceB.c	SourceC.c	SourceD.c	HeaderFile.h
Req.1	During recovery from brownout condition, navi plays from the beginning of the song. This specification is changed to play from the last position at low voltage detection.							
Reason	If navi has already created the song list, it can play from the last position at the time of recovering from brownout condition. So it's desired to unify action regardless of the timing.							
Comment	•brownout condition navi transfers to brownout condition by low voltage detection. Software is reseted at this timing.							
<Change the process of checking backup data>								
Branch Requirement	Req.1-1	As a result of integrity check backup data, change the return value from "disabled", "enabled" to "disabled", "Enabled" and "valid only for size".						
	Reason	Because the size in backup data is valid, the device is considered to same at the time of low voltage detection, and can play from the last position.						
	Comment	Consistency check backup data, the size of the devices (total size and free size) to compare. If size information is consistent with the backup data, it is determined the same as last connected device.						
Specification	<Change the size of device information acquisition process>							
		No change						
	<Change the size information comparison process>							
	Req.1-1-1	Add the definition of "valid only for size".						Def A
Req.1-1-2	Change the condition for clearing the backup data from not "OK" to not "valid backup" and "valid only for size".		Fun cA				Fun cD	
<Change the conditions of creating song list>								
Branch Requirement	Req.2-1	After integrity check backup data, change the timing of creating song list from only "invalid backup data" to "invalid backup data" and "valid for size".						
	Reason	Because navi must create the song list in case of "valid for size".						
	Comment	-						
Specification	Req.2-1-1	Backup data check results are "successful" Otherwise, navi create track list.						Fun cB
	Req.2-1-2	Song list is "not created successfully", navi create track list.						Fun cC

Whole Picture of Change Requirement Specifications and TM

Change Requirement Specifications



[Example] Change Design Documents

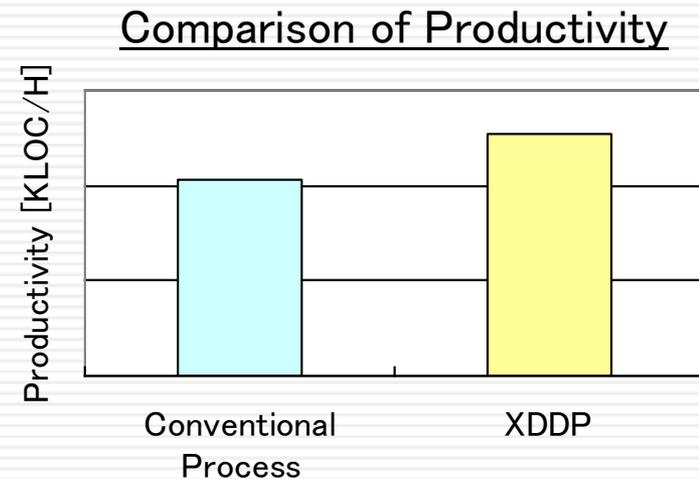
Project	XXX	Date	31/08/2011	
Source Name/Task Name	SourceFileA/ Module A	Author	XXX	
Change Requirement Specification	Change the condition for clearing the backup data from not "OK" to not "valid backup" and "valid only for size".	Modifier	XXX	<input type="checkbox"/> Modify
#Req.1-1-2		estimate lines	7	estimate time 1H
		change lines		actual time
<p><input type="checkbox"/> Policy of modify Nothing special.</p> <p><input type="checkbox"/> Change about structure of data No Change.</p> <p><input type="checkbox"/> Change about structure of function call The return value of FunctionB() becomes to "valid", "size only valid", and "error".</p> <p><input type="checkbox"/> Change about out of function</p>				
Item#	Change points		estimate lines	
1	Add the definition of "valid backup" and "valid only for size".		2	<input type="checkbox"/>
<p><input type="checkbox"/> Change about function</p>				
function	FunctionA()		<input checked="" type="checkbox"/> Chang <input type="checkbox"/> Add <input type="checkbox"/> Delete	
<p>Chage point</p>				
Item#	Change points		estimate time	
1	Change the timing of finishing process from not "OK" to not "valid backup" as the return value of FunctionB().		5	<input type="checkbox"/>

We applied XDDP to our navigation software development

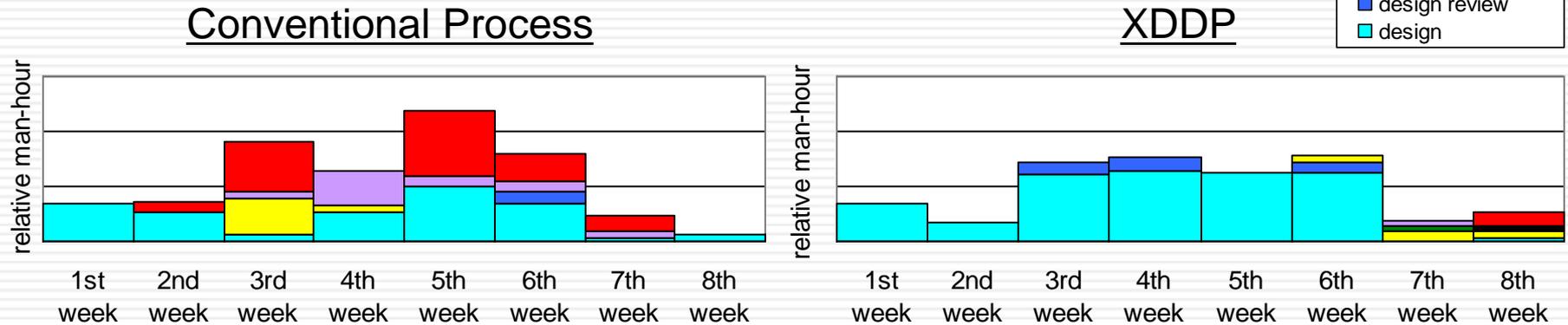
Case 1. General Project

Case 2. PWAT (2 types)

- Applied project
 - Prototype development
 - Development period : 2 months
 - Change size : 1,000 LOC (Line of Code)
- Result
 - Defects were decreased (from 2 to 0) in QA test
 - Productivity was increased 1.26 times



- Man-hours distribution



- Many defects were detected in test
Design, coding, and test were repeated

- Most of time was spent in design
Coding was finished at once

- Man-hours could be decreased although much time was spent in design

Change points were identified properly in XDDP

- Overall productivity can be improved even if we invest much time in identifying all change points
 - Hasty change of source code causes more work hours
 - Concrete change points (change specifications) make time in changing source code shorter

- Applied project

	PWAT(A)	PWAT(B)
Development Process	X-PWAT(A)	X-PWAT(B)
Target	Middleware (Audio Control)	Middleware (Voice Recognition)
Change Size [LOC]	300	500
Base Size [LOC]	26,000	9,500
Period [month]	1	3
Situation	Outsourced company developed base software	Other company developed base software

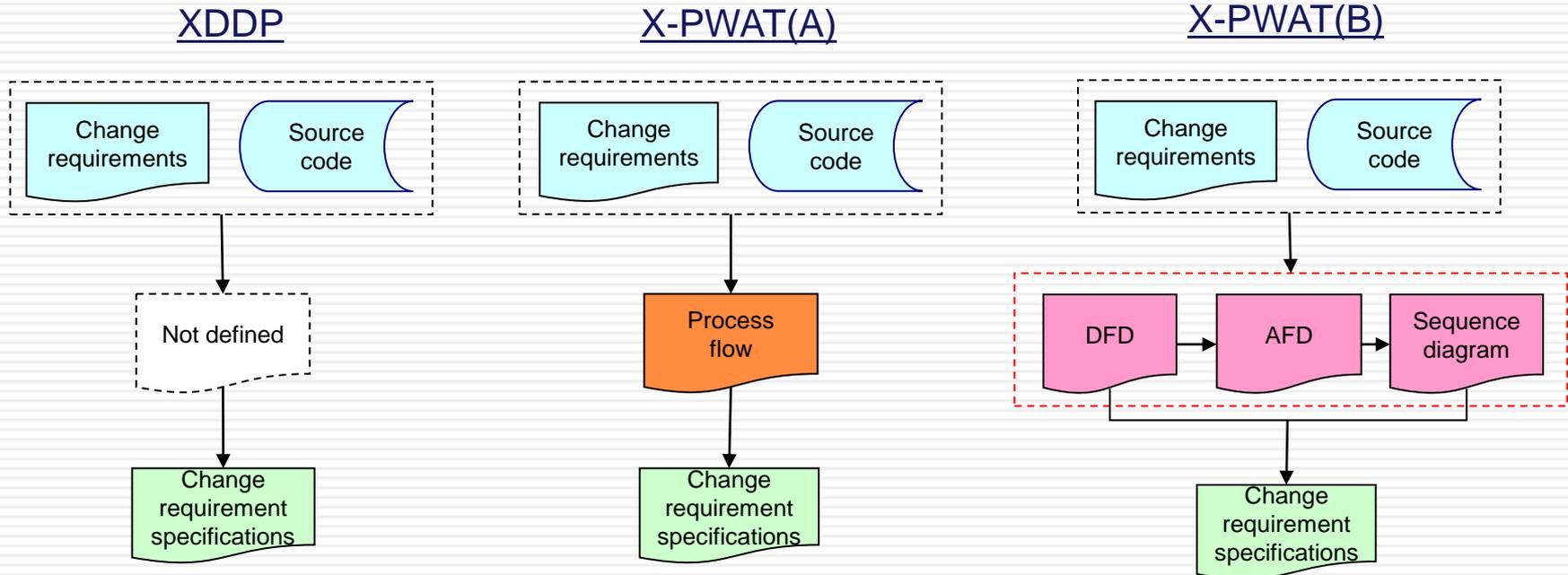
Apply XDDP to these two PWAT

- Policy
 - Investigation process should be designed in accordance with the knowledge level of engineer
- Reason
 - Investigation process is not defined in XDDP
 - Situation of project are not always the same

Design investigation Process by types of PWAT

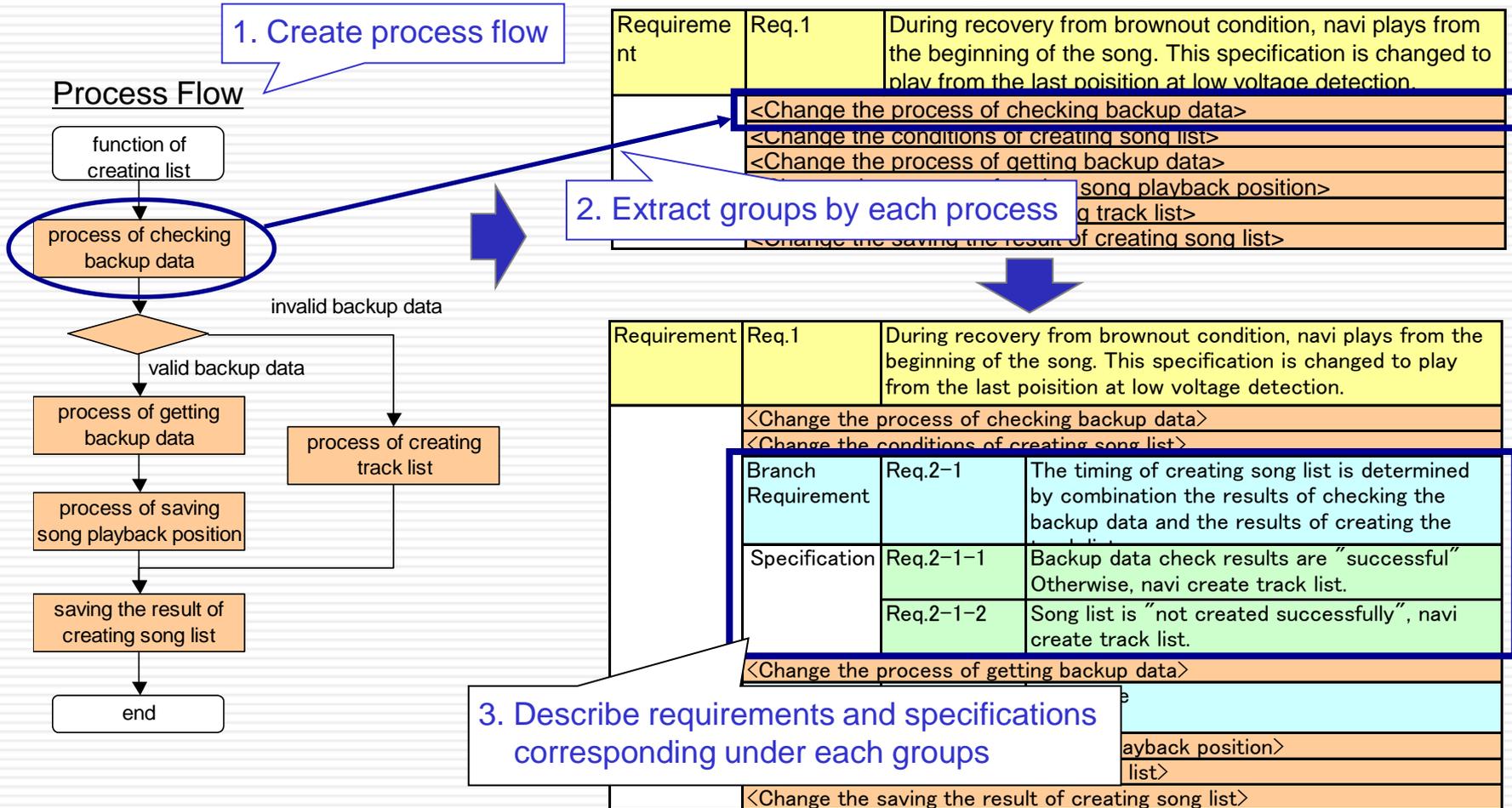
- Procedure
 1. Select software engineering techniques for investigations by types of PWAT
 2. Incorporate these techniques into XDDP
 - Define outcomes of investigation as input to make “Change Requirement Specifications”
- PWAT(A) / PWAT(B)
 - PWAT(A) : No information about the source code
 - Engineer can imagine function behavior
 - PWAT(B) : PWAT(A) + No knowledge of the domain
 - Engineer are not familiar with functions

- Outcomes of investigation
 - PWAT(A) : Process flow
 - PWAT(B) : DFD, AFD, Sequence Diagram



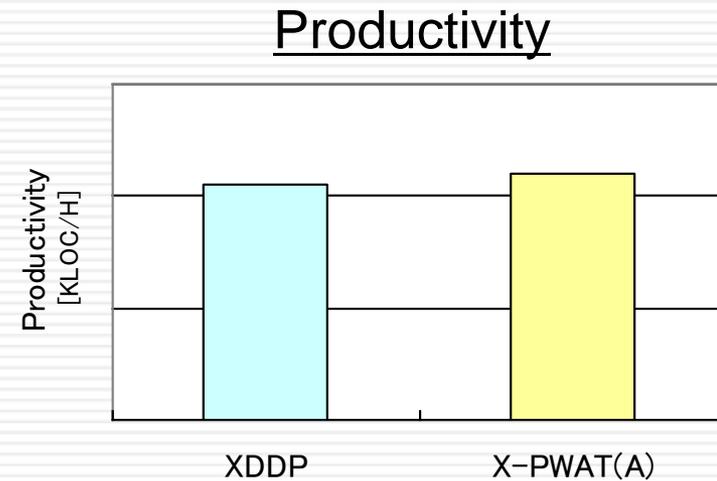
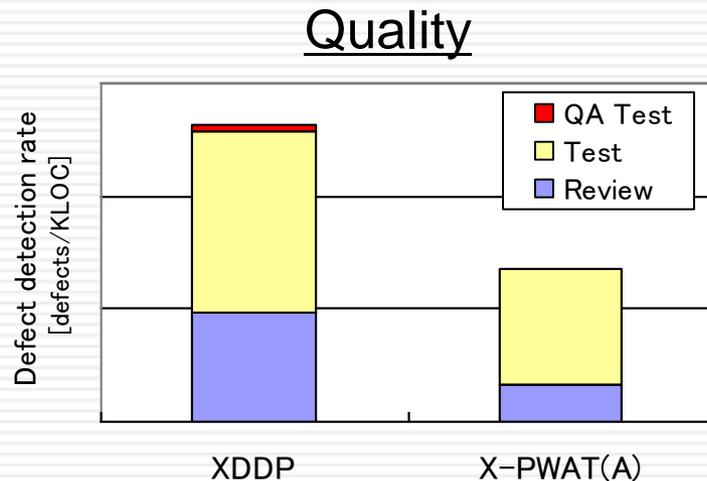
	PWAT(A)	PWAT(B)
Development Process	X-PWAT(A)	X-PWAT(B)
Target	Middleware (Audio Control)	Middleware (Voice Recognition)
Change Size [LOC]	300	500
Base Size [LOC]	26,000	9,500
Period [month]	1	3
Situation	Outsourced company developed base software	Other company developed base software

Change Requirement Specifications



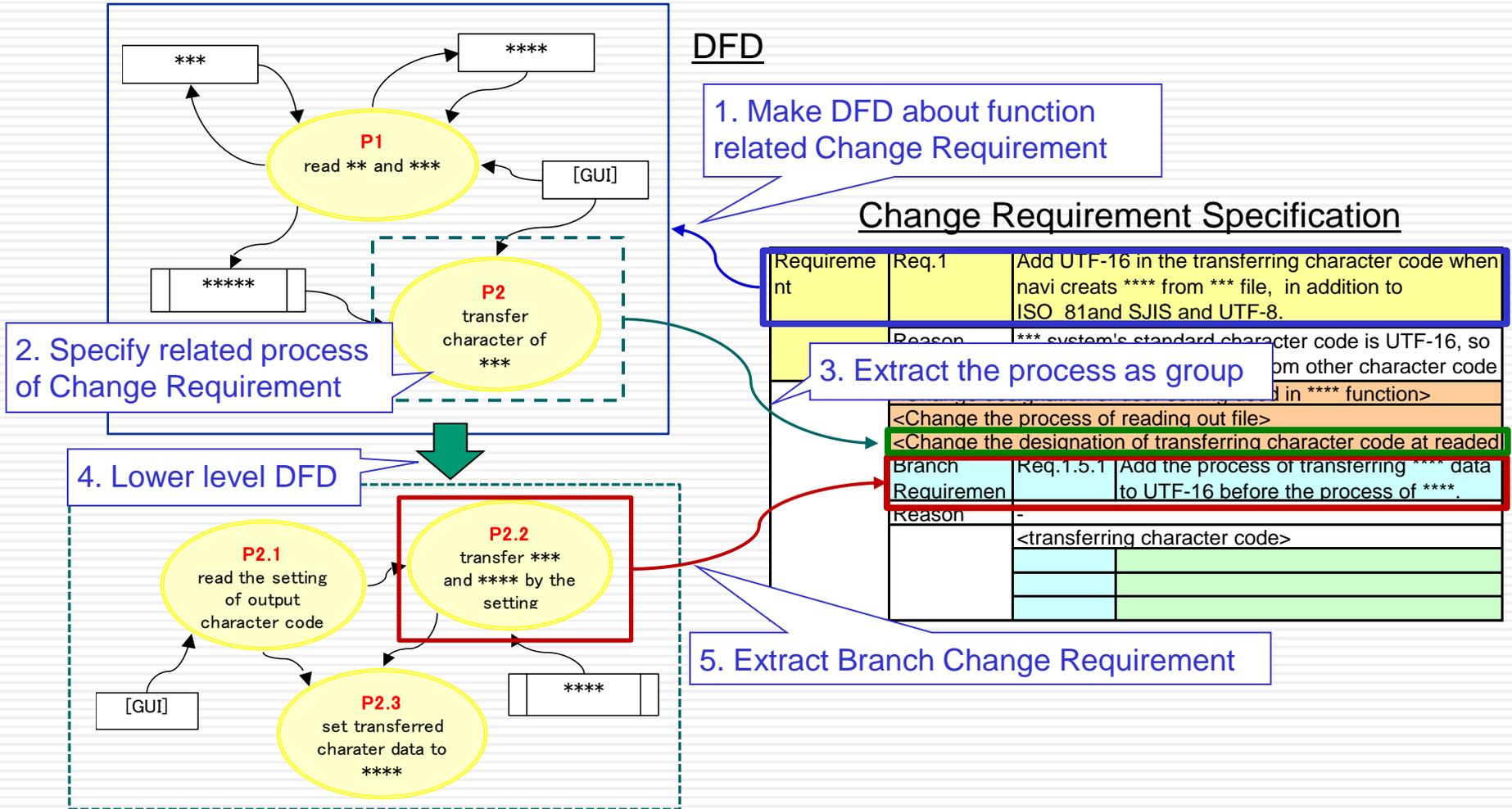
Detect Change Points by Process Flow

- Process
 - Compare X-PWAT(A) to simple XDDP
- Result

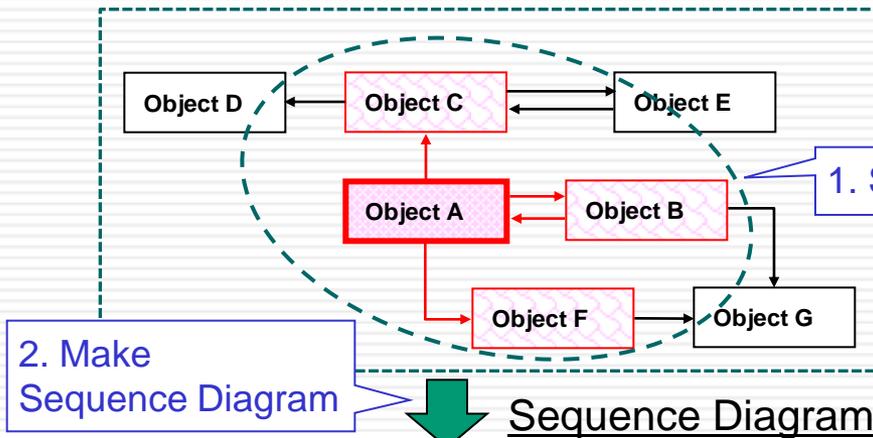


- Defect detection rate was decreased to about one-half
- Productivity was almost the same

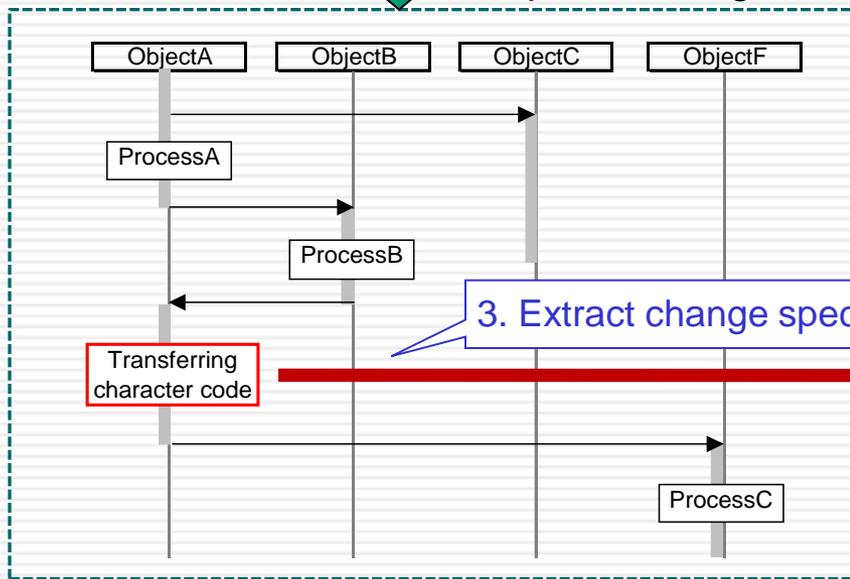
	PWAT(A)	PWAT(B)
Development Process	X-PWAT(A)	X-PWAT(B)
Target	Middleware (Audio Control)	Middleware (Voice Recognition)
Change Size [LOC]	300	500
Base Size [LOC]	26,000	9,500
Period [month]	1	3
Situation	Outsourced company developed base software	Other company developed base software



Identity the Change Process by DFD



Sequence Diagram

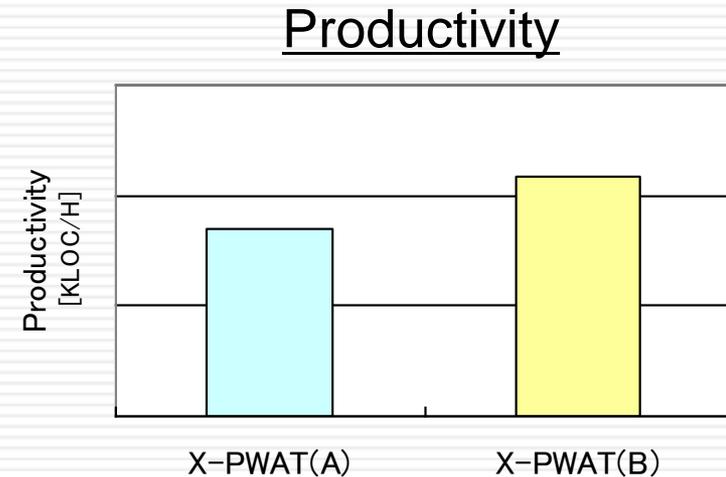
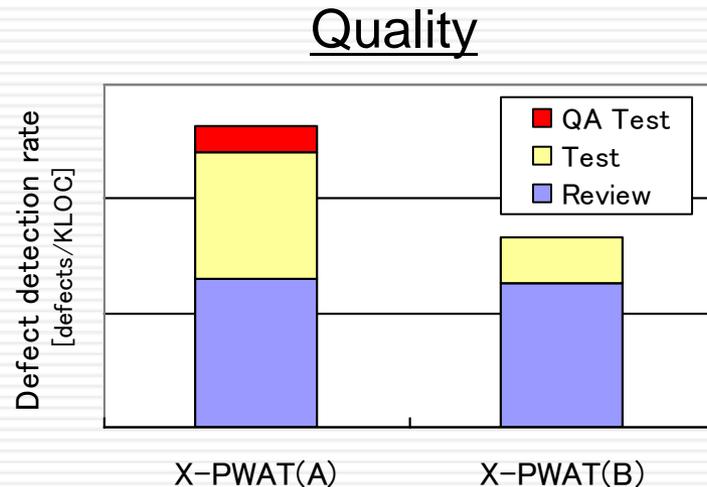


Change Requirement Specifications

Requirement	Req.1	Add UTF-16 in the transferring character code when navi creates **** from *** file, in addition to ISO_81and SJIS and UTF-8.
	Reason	*** system's standard character code is UTF-16, so this module must transfer from other character code
		<Change designation of user setting used in **** function>
		<Change the process of reading out file>
		<Change the designation of transferring character code at readed
Branch Requirement	Req.1.5.1	Add the process of transferring **** data to UTF-16 before the process of ****.
	Reason	-
		<transferring character code>
	Req.1.5.1.1	Add the process of definition and initialization of variables used in **** function before it is called.
	Req.1.5.1.2	After Req.1.5.1.1, get the data size of ****, in case it is transferred to UTF-16.
	Req.1.5.1.3	transfer the *** data to UTF-16, and reserve it to the area acquired at

Extract Change Specifications by Stepwise Refinement

- Process
 - Compare X-PWAT(B) to X-PWAT(A)
- Result



- Defect detection rate in QA Test was improved
- Productivity was improved by minimum necessary investigation

- It is important to design investigation process in XDDP
 - X-PWAT(A) was not effective to PWAT(B)
 - XDDP will be more effective by designing investigation process
- It is necessary to detect and describe specifications properly
 - In PWAT, it is difficult to understand all the base source code
 - We should make efforts to detect change specifications and describe them. That makes reviews more effective

- We applied XDDP to car navigation software development
 - Quality and productivity were improved
 - Overall productivity can be improved even if we spend much time in identifying all change points
- We confirmed XDDP could be applied to PWAT
 - We improved investigation process by designing the process in accordance with the knowledge level of engineer
 - The effectiveness was confirmed in our projects