

ທ້າວ ນຸຊົ່ວ ເຮີ ຫ້ອງ 3CW1

ສຶກສາບົດທີ 2 ແລ້ວໃຫ້ສະຫລຸບເນື້ອໃນຂອງບົດຮຽນດັ່ງກ່າວສົ່ງໃຫ້ອາຈານ

ບົດທີ 2

ຂະບວນການຜະລິດຊອບແວຣ໌

1. ຂະບວນການຜະລິດຊອບແວຣ໌

- ຂະບວນການ (Process)
 - ❖ ແມ່ນກຸ່ມຂອງຂັ້ນຕອນການເຮັດວຽກທີ່ປະກອບໄປດ້ວຍບັນດາກິດຈະກຳ, ຂໍ້ຈຳກັດ ຫຼື ເງື່ອນໄຂ ແລະ ຊັບພະຍາກອນຕ່າງໆທີ່ໃຊ້ຜະລິດໃຫ້ໄດ້ຜົນຮັບຕາມຕ້ອງການ
- ລັກສະນະຂອງຂະບວນການໂດຍທົ່ວໄປ
 - ❖ ຕ້ອງກຳນົດກິດຈະກຳທັງຫມົດຢ່າງຊັດເຈນ
 - ❖ ໃຊ້ຊັບພະຍາກອນພາຍໃຕ້ຂໍ້ຈຳກັດຕ່າງໆ ເພື່ອສ້າງຜະລິດຕະພັນ
 - ❖ ປະກອບຂຶ້ນຈາກຂະບວນການຍ່ອຍອື່ນໆທີ່ມີຄວາມສຳພັນກັນ
 - ❖ ທຸກກິດຈະກຳຈະຕ້ອງມີກຳນົດເລີ່ມຕົ້ນ ແລະ ສິ້ນສຸດ
 - ❖ ຕ້ອງມີເປົ້າຫມາຍຢ່າງຊັດເຈນ ແລະ ມີຫຼັກການໃນການປະຕິບັດ
 - ❖ ໃຊ້ຂໍ້ຈຳກັດໃນການຄວບຄຸມການດຳເນີນກິດຈະກຳ, ການໃຊ້ສັບພະຍາກອນຫຼືຜະລິດຕະພັນ
- ຂະບວນການຜະລິດຊອບແວຣ໌(Software Process)
 - ❖ ແມ່ນກຸ່ມຂອງກິດຈະກຳທີ່ກ່ຽວຂ້ອງກັນໃນການຜະລິດຊອບແວຣ໌ໃຫ້ໄດ້ຄຸນນະພາບ
 - ❖ ຂະບວນການທີ່ມີການກຳນົດລຳດັບຂັ້ນຕອນການເຮັດວຽກໄວ້ຢ່າງຊັດເຈນ ແລະ ຊອດຄ່ອງກັນຈະສາມາດນຳໄປໃຊ້ໃຫ້ເກີດປະໂຫຍດໄດ້
 - ❖ ຂະບວນການຊ່ວຍໃຫ້ສາມາດເຂົ້າໃຈ, ພິຈາລະນາ, ຄວບຄຸມ ແລະ ປັບປຸງກິດຈະກຳຕ່າງໆໃຫ້ເໝາະສົມ

2. ຄວາມສຳພັນລະຫວ່າງຂະບວນການຜະລິດຊອບແວຣ໌ ແລະ ຂະບວນການວິສະວະກຳຊອບແວຣ໌

- ຂະບວນການຜະລິດຊອບແວເປັນການດຳເນີນການຜະລິດຊອບແວຣ໌ທີ່ຕ້ອງໄດ້ຮັບການປະຕິບັດຕາມຫລັກການວິສະວະກຳຊອບແວຣ໌ເພື່ອໃຫ້ໄດ້ຊອບແວຣ໌ທີ່ມີຄຸນນະພາບ
- ວິສະວະກຳຊອບແວຣ໌ໄດ້ລວມເຖິງການສ້າງຂະບວນການ, ການປະເມີນ, ການຈັດການ, ການປ່ຽນແປງ, ການປັບປຸງຂະບວນການຜະລິດຊອບແວຣ໌ທີ່ຈະນຳມາໃຊ້ໃຫ້ມີປະສິດທິພາບ
- ເທັກໂນໂລຢີຂອງວິສະວະກຳຊອບແວຣ໌
 - ❖ Process ເປັນໂຄງສ້າງພື້ນຖານຂອງການດຳເນີນງານ
 - ❖ ເລືອກ Method ໃຫ້ເໝາະສົມແຕ່ລະຂັ້ນຕອນ
 - ❖ ເລືອກ Tool ແມ່ນເບິ່ງຈາກ Process ແລະ Method
- Process
 - ❖ ແມ່ນບັນດາຂັ້ນຕອນການເຮັດວຽກ ທີ່ປະກອບດ້ວຍບັນດາກິດຈະກຳ, ຂໍ້ຈຳກັດ ແລະ ຊັບພະຍາກອນທີ່ໃຊ້ໃນການຜະລິດ

3. ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

- Water fall Model
- Evolution Model (Iterative)
- Linear Sequential Model
- Prototype Model
- Rapid Application Development Model
- Incremental Model
- Boehm Spiral Model
- Rational Unified Process
- Component-Based Software Engineering
- Agile Process
- Extreme Programming

1) Water fall Model

ປະກອບດ້ວຍຂັ້ນຕອນການດຳເນີນງານຈັດລຽງຕໍ່ເນື່ອງກັນຕາມລຳດັບຂັ້ນຕອນຕໍ່ໄປຈະຖືກປະຕິບັດກໍ່ຕໍ່ເມື່ອຂັ້ນຕອນກ່ອນໜ້ານີ້ສຳເລັດ ປະກອບດ້ວຍ 5 ຂັ້ນຕອນພື້ນຖານ:

1. Requirement Definition
 2. System and Software Design
 3. Implementation and Unit Testing
 4. Integration and System Testing
 5. Operation
- ຈຸດດີ
 - ❖ ບໍ່ຍຸ້ງຍາກ ແລະ ໃຊ້ງານງ່າຍ
 - ❖ ຈັດການໄດ້ງ່າຍເນື່ອງມີຮູບແບບທີ່ແນ່ນອນ
 - ❖ ແຕ່ລະຂັ້ນຕອນຈະຖືກປະຕິບັດໃຫ້ສຳເລັດໃນເວລາທີ່ແນ່ນອນ
 - ❖ ໃຊ້ໄດ້ຜົນດີສຳຫລັບໂຄງການນ້ອຍທີ່ເຫັນຄວາມຕ້ອງການຢ່າງຊັດເຈນ
 - ຈຸດອ່ອນ
 - ❖ ບໍ່ສາມາດປ່ຽນແປງຂອບເຂດຂອງໂຄງການໃນລະຫວ່າງດຳເນີນງານ
 - ❖ ຈົນເຖິງຂັ້ນຕອນສຸດທ້າຍຊອບແວຣ໌ຈຶ່ງສາມາດໃຊ້ງານໄດ້
 - ❖ ບໍ່ແນ່ນອນ ແລະ ມີຄວາມສ່ຽງສູງ
 - ❖ ທີມງານແລະນັກວິເຄາະຈະຕ້ອງມີປະສົບການ ແລະ ສຳນານສູງ

2) Evolution Model (Iterative)

ເຮັດວຽກແບບເປັນຮອບວຽນ ປະກອບດ້ວຍ 4 ຂັ້ນຕອນ:

- ❖ Requirement Analysis

- ❖ System Design
- ❖ Coding and Unit Testing
- ❖ Assessment

3) Linear Sequential Model

- ເປັນການເຊື່ອມຕໍ່ແຕ່ລະກິດຈະກຳເປັນເສັ້ນຊື່ຕາມລຳດັບ
- ງ່າຍຕໍ່ກຳນົດເງິນກິດຈະກຳຕ່າງໆ
- ຕ້ອງເສຍເວລາກັບໄປເລີ່ມຕົ້ນໃໝ່ຖ້າມີບາງຂັ້ນຕອນຜິດພາດ
- ຈະໃຊ້ໄດ້ຜົນດີກໍຕໍ່ເມື່ອມີການກຳນົດຄວາມຕ້ອງການຢ່າງຈະແຈ້ງ ແລະ ບໍ່ມີການປ່ຽນ

4) Prototype Model

ເປັນການເຮັດຕົ້ນແບບຂຶ້ນມາເພື່ອຕ້ອງການກຳນົດຄວາມຕ້ອງການຂອງ
ຜູ້ໃຊ້, ລູກຄ້າ ຫຼື ຜູ້ທີ່ກ່ຽວຂ້ອງ

5) Rapid Application Development Model

ເປັນການພັດທະນາຊອບແວແບບກ້າວກະໂດດ ຊຶ່ງເປັນການແບ່ງວຽກ
ອອກເປັນຫລາຍສ່ວນ ແລ້ວແບ່ງແຕ່ລະສ່ວນໃຫ້ແຕ່ລະທີມງານເຮັດ
ເມື່ອສຳເລັດກໍນຳເອົາບັນດາສ່ວນຕ່າງໆມາລວມໃສ່ກັນ

6) Incremental Model

ເປັນຂະບວນການຜະລິດຊອບແວທີ່ມີການເພີ່ມຄວາມຕ້ອງການທີ່ສະຫລັບຊັບຊ້ອນ ຫຼື ເພີ່ມປະລິມານ
ຫລາຍຂຶ້ນເລື້ອຍໆ

7) Boehm Spiral Model

ປະສົມປະສານແນວຄິດຂອງແບບຈຳລອງ LSM, RAD ແລະ INM

ເຂົ້າດ້ວຍກັນ ເລີ່ມຈາກຄວາມຕ້ອງການໜ້ອຍແລ້ວຂະຫຍາຍອອກໄປເລື້ອຍໆປະກອບດ້ວຍ 6 ຂັ້ນຕອນ:

- Analysis and Planning
- Risk Analysis
- System Engineering
- Development Engineering
- Evaluation Customer Interaction
- Modification/Change to next version

8) Component-Based Software Engineering

ເປັນການຜະລິດຊອບແວຈາກອົງປະກອບຕ່າງໆທີ່ໄດ້ເຮັດໄວ້ແລ້ວ ມາປະກອບເຂົ້າກັນເປັນຊອບແວ ຊຶ່ງເປັນຫຼັກການ
Reusable ລຸດຕິຜົນໃນການຜະລິດໄດ້ຫຼາຍຊອບແວທີ່ໄດ້ມີຄຸນນະພາບລຸດເວລາໃນການຜະລິດ ແລະ ການບຳ
ລຸງຮັກສາ ແບ່ງອອກເປັນ 2 ສ່ວນທີ່ໄດ້ເຮັດພ້ອມກັນໄປ

- Domain Engineering ເປັນສ່ວນສ້າງ component ທີ່ໃຊ້ຊ້າໄດ້
- Component-base Development ເປັນສ່ວນການພັດທະນາຊອບແວຮ່ຈາກ component ທີ່ໄດ້ສ້າງໄວ້

9) Agile Process

- ເປັນການແຕກສາຂາມາຈາກ RAD
- ເຮັດການຜະລິດຊອບແວຮ່ແບບເລັ່ງດ່ວນ
- ບັງຄັບໃຫ້ເຮັດຕາມຂັ້ນຕອນຢ່າງເຄັ່ງຄັດ

10) Extreme Programming

- ເຮັດຕາມຫຼັກການການພັດທະນາແບບ Iteration ແລະ Incremental Development
- ເປັນແບບຈຳລອງທີ່ໃຊ້ແນວທາງວັດຖຸເປັນຫຼັກ
- ມີ 4 ຂັ້ນຕອນ
 - ❖ ວ່າງແຜນ, ອອກແບບ, ຂຽນໂປຣແກຣມ, ແລະ ທົດສອບ

4. ການປັບປຸງຂະບວນການດ້ວຍ CMM

- ເປົ້າຫມາຍສຳຄັນຂອງວິສະວະກຳຊອບແວຮ່ແມ່ນການຜະລິດຊອບແວຮ່ໃຫ້ມີຄຸນນະພາບ
- ຄຸນນະພາບບໍ່ໄດ້ຂຶ້ນກັບຕົວຜະລິດຕະພັນຊອບແວຮ່ພຽງຢ່າງດຽວ ແຕ່ຍັງຂຶ້ນກັບຂະບວນການຜະລິດຊອບແວຮ່ທີ່ເລືອກໃຊ້ນຳອີກ
- ເມື່ອມີການນຳເທັກນິກ, ຫລັກການ ຫຼື ເຄື່ອງມື ຊະນິດໃໝ່ເຂົ້າມາປະຍຸກໃຊ້ກໍຈຳເປັນຈະຕ້ອງປັບຂະບວນການໃຫ້ສອດຄ່ອງກັບສິ່ງເລົ່ານັ້ນນຳ
- ຍຸດທະສາດໃນການປັບປຸງຂະບວນການນັ້ນມີຫລາຍຮູບແບບເຊັ່ນ: Management, Business Process, Six Sigma ເປັນຕົ້ນ
- ແບບຈຳລອງວຸດທິຄວາມສາມາດ(Capability Maturity Model)
- ແບບຈຳລອງວຸດທິຄວາມສຳມັດ (Capability Maturity Model)
- ຄຸນລັກສະນະ ແລະ ຂະບວນການສຳຄັນຂອງວຸດທິຄວາມສາມາດ
 - ❖ ລະດັບ 1
 - ຄຸນລັກສະນະ
 - ບໍ່ເປັນລະບຽບ
 - ບໍ່ສາມາດກະທຳຊ້າ ຫຼື ນຳມາໃຊ້ຄືນໄດ້
 - ມີຄວາມສ່ຽງສູງຫລາຍ
 - ຂະບວນການໃນແຕ່ລະດ້ານ
 - ບໍ່ມີການກຳຫນົດ
 - ❖ ລະດັບ 2
 - ຄຸນລັກສະນະ

- ມີນະໂຍບາຍຈະແຈ້ງ
- ສາມາດກະທຳຊ້າໄດ້
- ບໍ່ມີການປັບປຸງ

❖ ລະດັບ 2

- ຂະບວນການໃນແຕ່ລະດ້ານ
 - ການວິເຄາະຄວາມຕ້ອງການ
 - ການວາງແຜນໂຄງການ
 - ການປະກັນຄຸນນະພາບຊອບແວຣ໌
 - ການຊອກຫາຜູ້ປະຕິບັດຕໍ່ໄປ ຫຼື ຜູ້ຮັບຈ້າງ
 - ການຈັດສະພາບແວດລ້ອມຊອບແວຣ໌

❖ ລະດັບ 3

- ຄຸນລັກສະນະ
 - ມີການປັບປຸງປະສິດທິພາບໃນດ້ານຕ່າງໆເຊັ່ນ: ຕົ້ນທຶນ, ກຳນົດເວລາ, ຄຸນນະພາບ, ຄວາມສ່ຽງ
- ຂະບວນການໃນແຕ່ລະດ້ານ
 - ການຈັດການຂະບວນການດ້ວຍເອກະສານ
 - ການປັບປຸງໃຫມ່, ການຝຶກອົບຮົມ, ການຈັດການບຸກຄົນ
 - ການຈັດການຄຸນນະພາບເບື້ອງຕົ້ນ
 - ການສະໜັບສະໜູນການຜະລິດຊອບແວຣ໌

❖ ລະດັບ 4

- ຄຸນລັກສະນະ
 - ມີປະສິບການ
 - ມີການປັບປຸງປະສິທິພາບຢ່າງຕໍ່ເນື່ອງ
- ຂະບວນການໃນແຕ່ລະດ້ານ
 - ການຈັດການຂະບວນການທາງດ້ານປະລິມານ
 - ການຈັດການຄຸນນະພາບຊອບແວຣ໌

5. ວິທີການ (Methodology)

- Formal Methodology

❖ ການເຮັດວຽກມີ 2 ຊະນິດ

- Formal Specification ເປັນວິທີອະທິບາຍຂໍ້ກຳນົດຫຼັກການຄະນິດສາດໃດຫນຶ່ງເຊັ່ນ: ພຶດຊະຄະນິດ, ແບບຈຳລອງທາງຄະນິດສາດ
- Formal Verification ເປັນວິທີການກວດສອບໂດຍໃຊ້ຫຼັກການພິສູດທາງຕັກກະສາດ