

# 252SE311: ວິສະວະກຳຊອບແວ 2020-2021

ຄວາມຮູ້ເບື້ອງຕົ້ນກ່ຽວກັບວິສະວະກຳຊອບແວ

ບົດທີ 2

ຂະບວນການຜະລິດຊອບແວ

# ເນື້ອໃນຫຍໍ້



- ◆ ຂະບວນການຜະລິດຊອບແວຣ໌
- ◆ ຄວາມສໍາພັນລະຫວ່າງຂະບວນການຜະລິດຊອບແວຣ໌ ແລະ ຂະບວນການວິສະວະກໍາຊອບແວຣ໌
- ◆ ແບບຈໍາລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌
- ◆ ການປັບປຸງຂະບວນການດ້ວຍ CMM ແລະ CMMI
- ◆ ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກໍາຊອບແວຣ໌

# ຂະບວນການຜະລິດຊອບແວຣ໌

## ➤ ຂະບວນການ (Process)

- ແມ່ນກຸ່ມຂອງຂັ້ນຕອນການເຮັດວຽກ ທີ່ປະກອບໄປດ້ວຍບັນດາກິດຈະກຳ, ຂໍ້ຈຳກັດ ຫຼື ເງື່ອນໄຂ ແລະ ຊັບພະຍາກອນຕ່າງໆ ທີ່ຈະໃຊ້ຜະລິດໃຫ້ໄດ້ຜົນ ຮັບຕາມຕ້ອງການ

## ➤ ລັກສະນະຂອງຂະບວນການໂດຍທົ່ວໄປ

- ຕ້ອງກຳນົດກິດຈະກຳທັງໝົດຢ່າງຊັດເຈນ
- ໃຊ້ຊັບພະຍາກອນພາຍໃຕ້ຂໍ້ຈຳກັດຕ່າງໆ ເພື່ອສ້າງຜະລິດຕະພັນ
- ປະກອບຂຶ້ນຈາກຂະບວນການຍ່ອຍອື່ນໆທີ່ມີຄວາມສຳພັນກັນ
- ທຸກກິດຈະກຳຈະຕ້ອງມີການເລີ່ມຕົ້ນ ແລະ ສິ້ນສຸດ
- ຕ້ອງມີເປົ້າໝາຍຢ່າງຊັດເຈນ ແລະ ມີຫຼັກການໃນການປະຕິບັດ
- ໃຊ້ຂໍ້ຈຳກັດໃນການຄວບຄຸມການດຳເນີນກິດຈະກຳ, ການໃຊ້ສັບພະຍາກອນ ຫຼື ຜະລິດຕະພັນ

# ຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ ຂະບວນການຜະລິດຊອບແວຣ໌ (Software Process)

- ແມ່ນກຸ່ມຂອງກິດຈະກຳທີ່ກ່ຽວຂ້ອງກັນໃນການຜະລິດຊອບແວຣ໌ໃຫ້ໄດ້ຄຸນນະພາບ
- ຂະບວນການທີ່ມີການກຳນົດລຳດັບຂັ້ນຕອນການເຮັດວຽກໄວ້ຢ່າງຊັດເຈນ ແລະ ຊອດຄ່ອງກັນຈະສາມາດນຳໄປໃຊ້ໃຫ້ເກີດປະໂຫຍດໄດ້
- ຂະບວນການຊ່ວຍໃຫ້ສາມາດເຂົ້າໃຈ, ພິຈາລະນາ, ຄວບຄຸມ ແລະ ປັບປຸງກິດຈະກຳຕ່າງໆໃຫ້ເໝາະສົມ

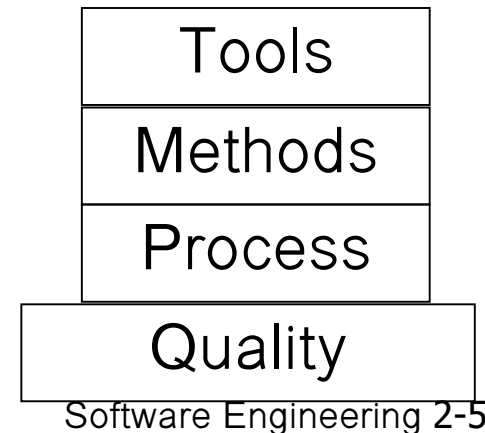
# ຄວາມສຳພັນລະຫວ່າງຂະບວນການຜະລິດຊອບແວ ແລະ ຂະບວນການວິສະວະກຳຊອບແວ

➤ ຂະບວນການຜະລິດຊອບແວເປັນການດຳເນີນການຜະລິດຊອບແວທີ່ຕ້ອງ  
ໄດ້ຮັບການປະຕິບັດຕາມຫຼັກການວິສະວະກຳຊອບແວເພື່ອໃຫ້ໄດ້ຊອບ  
ແວທີ່ມີຄຸນນະພາບ

■ ວິສະວະກຳຊອບແວໄດ້ລວມເຖິງການສ້າງຂະບວນການ, ການປະເມີນ,  
ການຈັດການ, ການປ່ຽນແປງ, ການປັບປຸງຂະບວນການຜະລິດຊອບແວ  
ທີ່ຈະນຳມາໃຊ້ໃຫ້ມີປະສິດທິພາບ

➤ ເທັກໂນໂລຍີຂອງວິສະວະກຳຊອບແວ

- Process ເປັນໂຄງສ້າງພື້ນຖານຂອງການດຳເນີນງານ
- ເລືອກ Method ໃຫ້ເໝາະສົມແຕ່ລະຂັ້ນຕອນ
- ເລືອກ Tool ແມ່ນເບິ່ງຈາກ Process ແລະ Method



# ຄວາມສໍາພັນລະຫວ່າງຂະບວນການຜະລິດຊອບແວຣ໌ ແລະ ຂະບວນການວິສະວະກໍາຊອບແວຣ໌


## ↳ Process

- ແມ່ນບັນດາຂັ້ນຕອນການເຮັດວຽກ ທີ່ປະກອບດ້ວຍບັນດາກິດຈະກຳ, ຂໍ້ຈຳກັດ ແລະ ຊັບພະຍາກອນທີ່ໃຊ້ໃນການຜະລິດ
- ໂດຍທົ່ວໄປຈະມີລັກສະນະດັ່ງນີ້
  - ກຳໜົດກິດຈະກຳທັງໝົດຢ່າງຊັດເຈນ
  - ໃຊ້ຊັບພະຍາກອນພາຍໃຕ້ຂໍ້ຈຳກັດ (ເວລາ ແລະ ຕົ້ນທຶນ)
  - ອາດຈະປະກອບດ້ວຍບັນດາຂະບວນການຍ່ອຍອື່ນໆ
  - ມີເງື່ອນໄຂໃນການເລີ່ມຕົ້ນ ແລະ ສິ້ນສຸດ
  - ທຸກຂັ້ນຕອນ, ທຸກກິດຈະກຳຈະຕ້ອງມີເປົ້າໝາຍຊັດເຈນ
  - ຂໍ້ຈຳກັດຫຼືເງື່ອນໄຂສາມາດເອົາໃຊ້ຄວບຄຸມການດຳເນີນກິດຈະກຳ, ການໃຊ້ຊ/

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວ

- ↳ 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

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌



## ↳ Water fall Model

- ປະກອບດ້ວຍຂັ້ນຕອນການດຳເນີນງານຈັດລຽງຕໍ່ເນື່ອງກັນຕາມລຳດັບ
- ຂັ້ນຕອນຕໍ່ໄປຈະຖືກປະຕິບັດກໍ່ຕໍ່ເມື່ອຂັ້ນຕອນກ່ອນໜ້ານີ້ສຳເລັດ
- ປະກອບດ້ວຍ 5 ຂັ້ນຕອນພື້ນຖານ (ຮູບແຕ້ມໃນສະໄລ່ຕໍ່ໄປ)
  - Requirement Definition
  - System and Software Design
  - Implementation and Unit Testing
  - Integration and System Testing
  - Operation and Maintenance



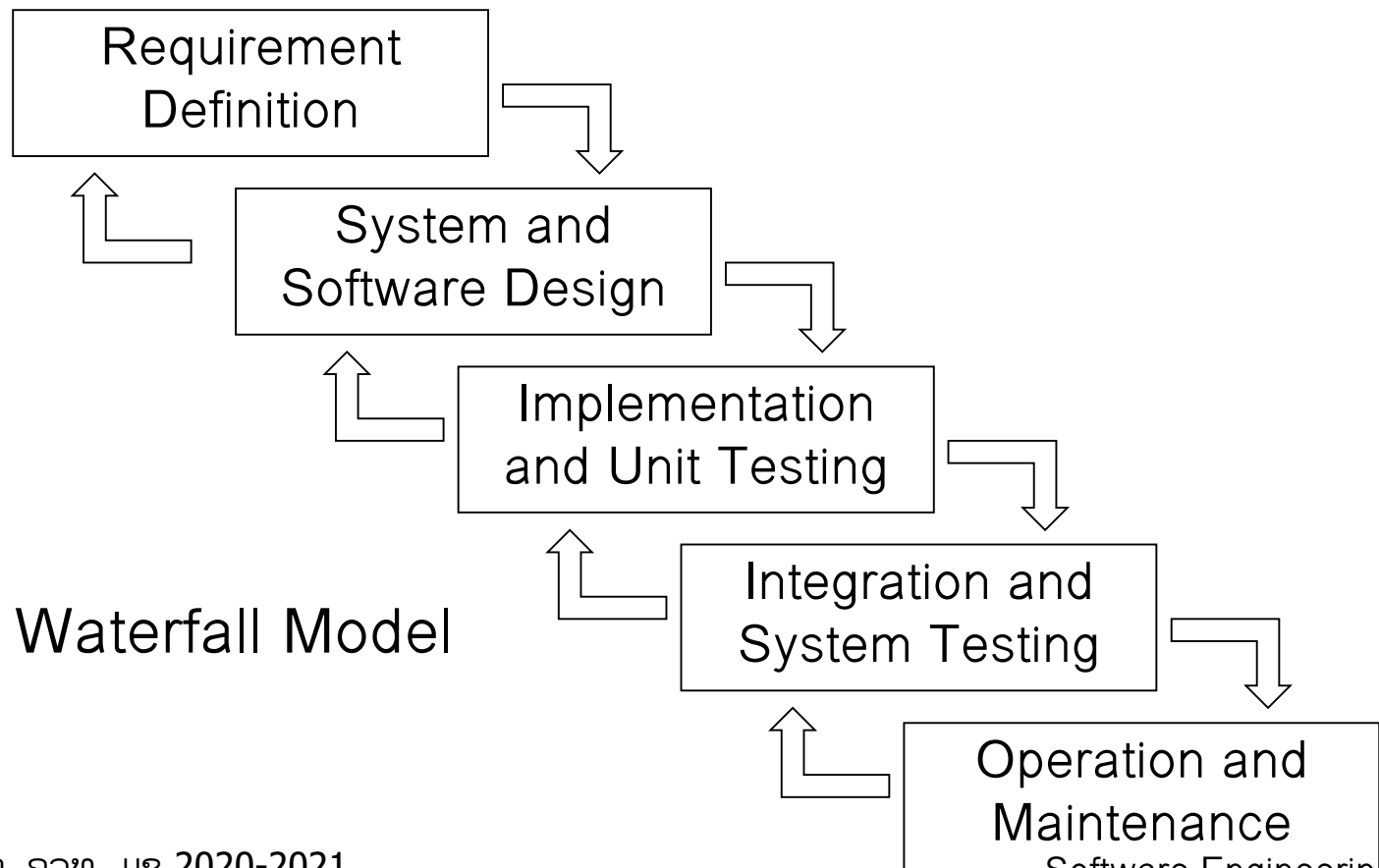
# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ Water fall Model

- ຈຸດດີ
  - ບໍ່ຍຸ້ງຍາກ ແລະ ໃຊ້ງານງ່າຍ
  - ຈັດການໄດ້ງ່າຍເນື່ອງມີຮູບແບບທີ່ແນ່ນອນ
  - ແຕ່ລະຂັ້ນຕອນຈະຖືກປະຕິບັດໃຫ້ສຳເລັດໃນເວລາທີ່ແນ່ນອນ
  - ໃຊ້ໄດ້ຜົນດີສຳຫລັບໂຄງການນ້ອຍທີ່ເຫັນຄວາມຕ້ອງການຢ່າງຊັດເຈນ
- ຈຸດອ່ອນ
  - ບໍ່ສາມາດປ່ຽນແປງຂອບເຂດຂອງໂຄງການໃນລະຫວ່າງດຳເນີນງານ
  - ຈົນເຖິງຂັ້ນຕອນສຸດທ້າຍຊອບແວຣ໌ຈຶ່ງສາມາດໃຊ້ງານໄດ້
  - ບໍ່ແນ່ນອນ ແລະ ມີຄວາມສ່ຽງສູງ
  - ທີມງານແລະນັກວິເຄາະຈະຕ້ອງມີປະສົບການ ແລະ ສຳນານສູງ


# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

↘ Water fall Model (ປະກອບດ້ວຍ 5 ຂັ້ນຕອນ)



Adaptive Waterfall Model

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

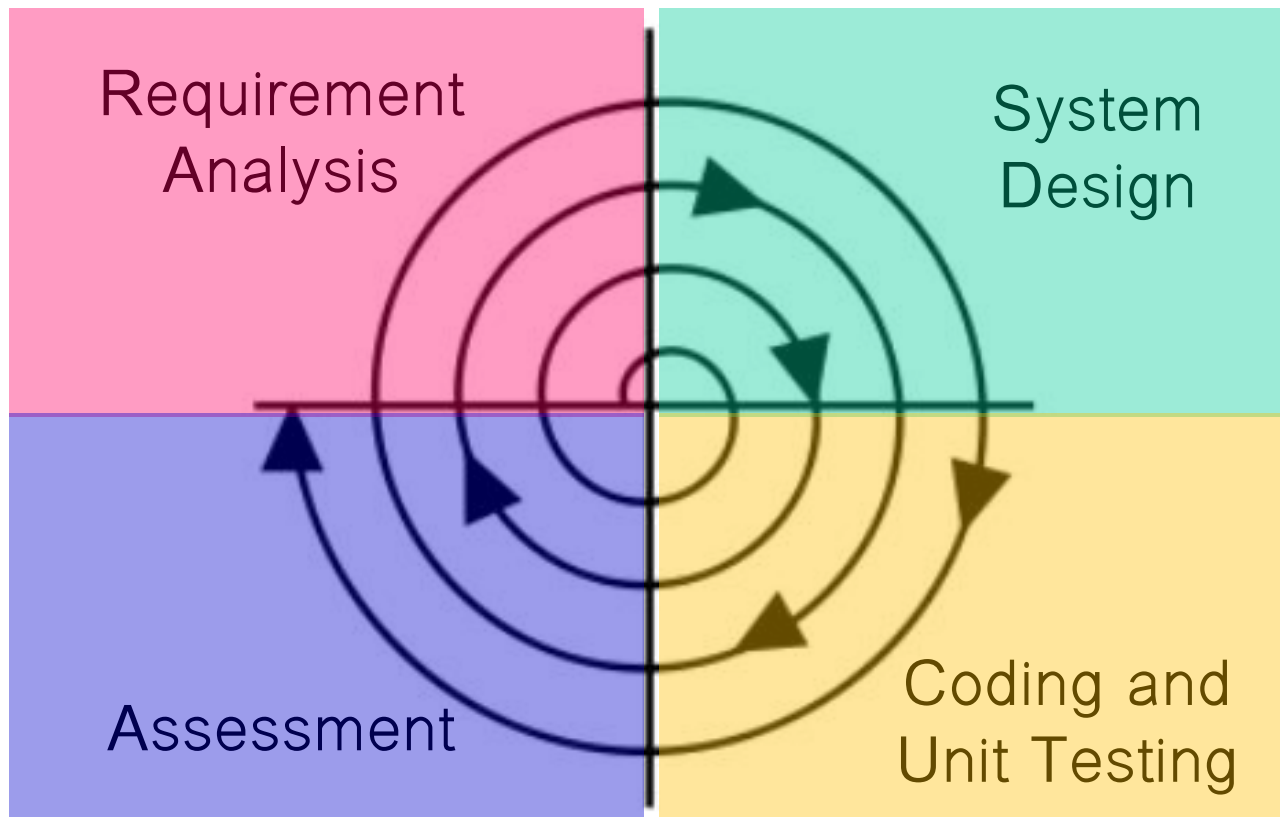


## ↳ Evolution Model (Iterative)

- ເຮັດວຽກແບບເປັນຮອບວຽນ
- ປະກອບດ້ວຍ 4 ຂັ້ນຕອນ
  - Requirement Analysis
  - System Design
  - Coding and Unit Testing
  - Assessment

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

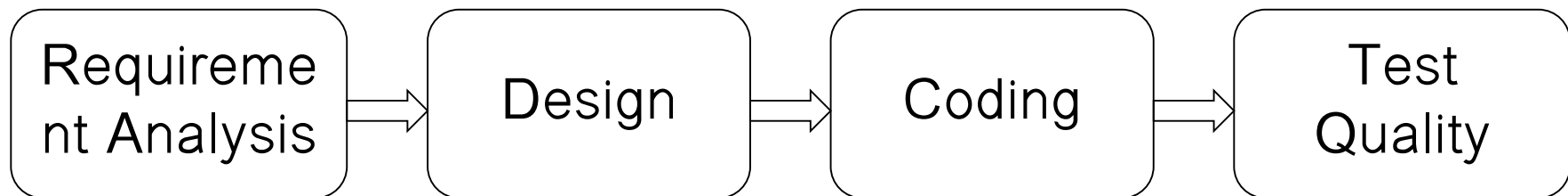
↳ Evolution Model (Iterative)



# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ Linear Sequential Model

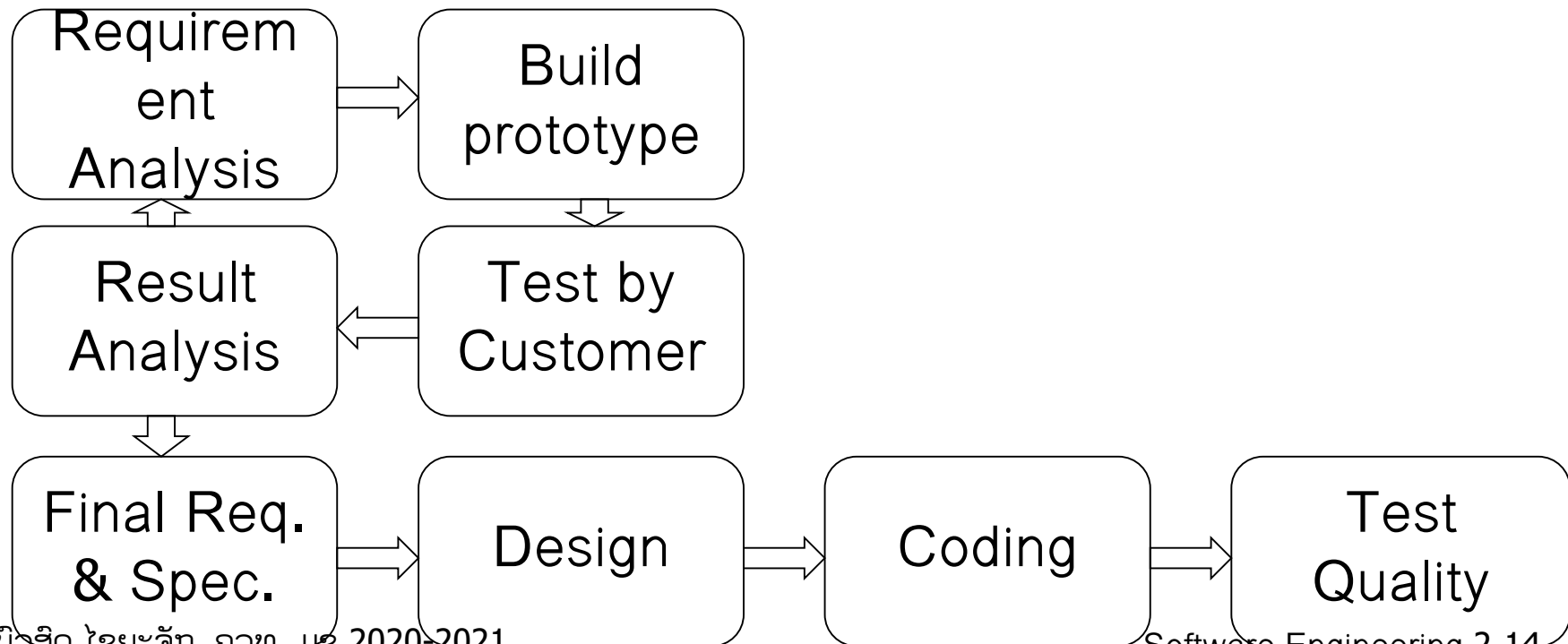
- ເປັນການເຊື່ອມຕໍ່ແຕ່ລະກິດຈະກຳເປັນເສັ້ນຊື່ຕາມລຳດັບ
- ງ່າຍຕໍ່ການດຳເນີນກິດຈະກຳຕ່າງໆ
- ຕ້ອງເສຍເວລາກັບໄປເລີ່ມຕົ້ນໃໝ່ຖ້າມີບາງຂັ້ນຕອນຜິດພາດ
- ຈະໃຊ້ໄດ້ຜົນດີກໍຕໍ່ເມື່ອມີການກຳນົດຄວາມຕ້ອງການຢ່າງຈະແຈ້ງ ແລະ ບໍ່ມີການປ່ຽນ




# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ Prototype Model

- ເປັນການເຮັດຕົ້ນແບບຂຶ້ນມາເພື່ອຕ້ອງການກຳນົດຄວາມຕ້ອງການຂອງຜູ້ໃຊ້, ລຸກຄ້າ ຫຼື ຜູ້ທີ່ກ່ຽວຂ້ອງ



# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌



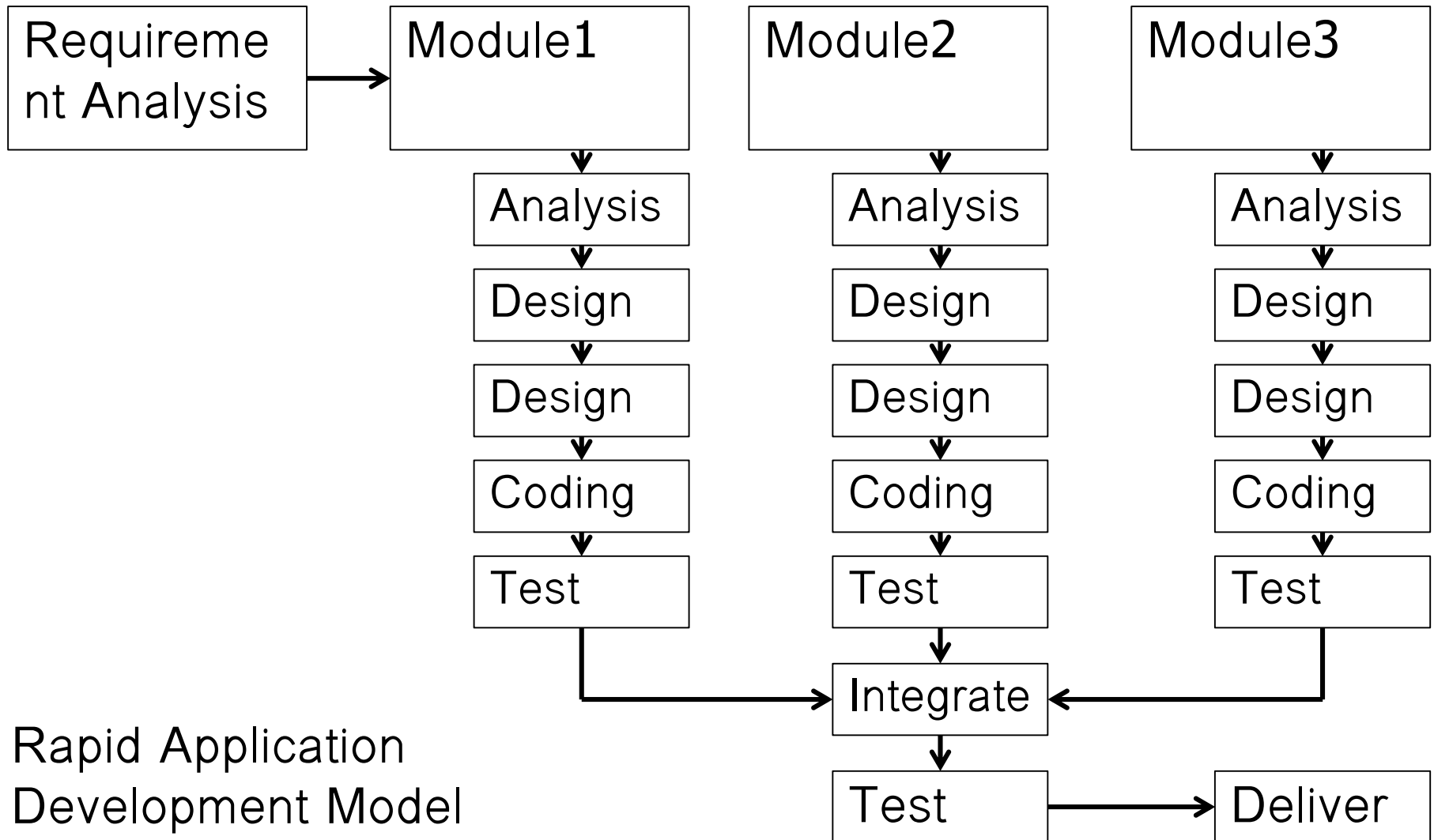
## ↳ Rapid Application Development Model

- ເປັນການພັດທະນາຊອບແວຣ໌ແບບກ້າວກະໂດດ ຊຶ່ງເປັນການແບ່ງວຽກອອກເປັນຫລາຍສ່ວນ ແລ້ວແບ່ງແຕ່ລະສ່ວນໃຫ້ແຕ່ລະທີມງານເຮັດ ເມື່ອສຳເລັດກໍນຳເອົາບັນດາສ່ວນຕ່າງໆມາລວມໃສ່ກັນ

## ↳ Incremental Model


- ເປັນຂະບວນການຜະລິດຊອບແວຣ໌ທີ່ມີການເພີ່ມຄວາມຕ້ອງການທີ່ສະຫລັບຊັບຊ້ອນ ຫຼື ເພີ່ມປະລິມານຫລາຍຂຶ້ນເລື້ອຍໆ

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌





# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌



## ↳ Boehm Spiral Model

- ປະສົມປະສານແນວຄິດຂອງແບບຈຳລອງ LSM, RAD ແລະ INM ເຂົ້າດ້ວຍກັນ
- ເລີ່ມຈາກຄວາມຕ້ອງການໜ້ອຍແລ້ວຂະຫຍາຍອອກໄປເລື້ອຍໆ
- ປະກອບດ້ວຍ 6 ຂັ້ນຕອນ
  - Analysis and Planning
  - Risk Analysis
  - System Engineering
  - Development Engineering
  - Evaluation Customer Interaction
  - Modification/Change to next version

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ➤ Boehm Spiral Model

- ຈຸດດີ
  - ມີຄວາມຢືດຢູ່ນສູງ
  - ສາມາດຂ້າມຂັ້ນຕອນໄດ້ຖ້າບໍ່ມີຄວາມຈຳເປັນໃນບາງຮອບ
  - ເໝາະສົມກັບລະບົບທີ່ມີການປ່ຽນແປງຄວາມຕ້ອງການເປັນປະຈຳ
- ຈຸດອ່ອນ
  - ມີຄວາມສ່ຽງສູງ
  - ຕ້ອງວິເຄາະຄວາມສ່ຽງທຸກຮອບ

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ Rational Unified Process

- ເປັນການປະສົມປະສານແບບຈຳລອງ Iteration, Evolution, Increment ເຂົ້າດ້ວຍກັນ
- ໄດ້ແບ່ງອອກເປັນ 4 phase
  - Inception
  - Elaboration
  - Construction
  - Transition
- ສາມາດສະແດງໃຫ້ເຫັນຂະບວນການຜະລິດຊອບແວຣ໌ໃນ 3 ຢ່າງ
  - Dynamic Perspective ສະແດງໃຫ້ເຫັນຂັ້ນຕອນການເຮັດວຽກ 4 ເຟສ
  - Static Perspective ສະແດງໃຫ້ເຫັນກິດຈະກຳທີ່ຕ້ອງເຮັດ
  - Practice Perspective ແນະນຳຫຼັກການປະຕິບັດງານທີ່ຕ້ອງໃຊ້ໃຈ ຂ/ບ

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

- ↳ Inception ເປັນໄລຍະເລີ່ມຕົ້ນໃນການປະຕິບັດງານ
  - ກຳນົດຂອບເຂດ, ໜ້າທີ່ການເຮັດວຽກຫຼັກ, ວິໄສທັດ
  - ເປົ້າໝາຍແມ່ນເພື່ອກຳນົດພາກສ່ວນພາຍນອກທີ່ປະຕິບັດຕໍ່ລະບົບ
- ↳ Elaboration
  - ເປົ້າໝາຍແມ່ນໃຫ້ເຂົ້າໃນຕໍ່ບັນຫາຂອງລະບົບ, ກຳນົດສະຖາປັດຕະຍະກຳຂອງລະບົບ, ເຮັດແຜນງານ, ຊອກຫາຄວາມສ່ຽງ
  - ສິ່ງທີ່ໄດ້ແມ່ນແບບຈຳລອງຄວາມຕ້ອງການຂອງລະບົບ (User Case Diagram)
- ↳ Construction
  - ເປັນສ່ວນທີ່ກ່ຽວຂ້ອງກັບການອອກແບບ, ຂຽນໂປຣແກຣມ ແລະ ທົດສອບ
  - ສິ່ງທີ່ໄດ້ແມ່ນຊອບແວຣ໌ ແລະ ເອກະສານຂອງຊອບແວຣ໌
- ↳ Transition
  - ສິ່ງມອບໃຫ້ກັບລູກຄ້າ, ຕິດຕັ້ງໃຊ້ງານ ແລະ ຝຶກອົບຮົມ

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວ

## ↳ ດ້ານຄຳແນະນຳການປະຕິບັດງານ

- ພັດທະນາຊອບແວແບບວິນຊໍາ
- ມີການບໍລິຫານຈັດການຄວາມຕ້ອງການຂອງຊອບແວ
- ໃຊ້ສະຖາປັດຕະຍະກຳແບບ Component
- ໃຊ້ແບບຈຳລອງທີ່ເປັນແຜ່ນພາບ
- ກວດສອບຄຸນນະພາບຂອງຊອບແວສະເໝີ
- ຄວບຄຸມການປ່ຽນແປງຂອງຊອບແວ


## ↳ ບໍ່ສາມາດເອົາໄປໃຊ້ໄດ້ກັບການພັດທະນາຊອບແວທຸກປະເພດ

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌

## ↳ Component-Based Software Engineering

- ເປັນການຜະລິດຊອບແວຣ໌ຈາກ ອົງປະກອບຕ່າງທີ່ໄດ້ເຮັດໄວ້ແລ້ວ ມາປະກອບເຂົ້າກັນເປັນຊອບແວຣ໌ ຊຶ່ງເປັນຫຼັກການ Reusable
- ລຸດຕິິນທິນໃນການຜະລິດໄດ້ຫຼາຍ
- ຊອບແວຣ໌ທີ່ໄດ້ມີຄຸນນະພາບ
- ລຸດເວລາໃນການຜະລິດ ແລະ ການບຳລຸງຮັກສາ
- ແບ່ງອອກເປັນ 2 ສ່ວນທີ່ໄດ້ເຮັດພ້ອມກັນໄປ
  - Domain Engineering ເປັນສ່ວນສ້າງ component ທີ່ໃຊ້ຊ້ຳໄດ້
  - Component-base Development ເປັນສ່ວນການພັດທະນາຊອບແວຣ໌ຈາກ component ທີ່ໄດ້ສ້າງໄວ້

# ແບບຈຳລອງຂອງຂະບວນການຜະລິດຊອບແວຣ໌



## ➤ Agile Process

- ເປັນການແຕກສາຂາມາຈາກ RAD
- ເຮັດການຜະລິດຊອບແວຣ໌ແບບເລັ່ງດ່ວນ
- ບັງຄັບໃຫ້ເຮັດຕາມຂັ້ນຕອນຢ່າງເຄັ່ງຄັດ

## ➤ Extreme Programming

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

# ການປັບປຸງຂະບວນການດ້ວຍ CMM

- ◆ ເປົ້າໝາຍສໍາຄັນຂອງວິສະວະກໍາຊອບແວຣ໌ແມ່ນການຜະລິດຊອບແວຣ໌ໃຫ້ມີຄຸນນະພາບ
- ◆ ຄຸນນະພາບບໍ່ໄດ້ຂຶ້ນກັບຕົວຜະລິດຕະພັນຊອບແວຣ໌ພຽງຢ່າງດຽວ ແຕ່ຍັງຂຶ້ນກັບຂະບວນການຜະລິດຊອບແວຣ໌ທີ່ເລືອກໃຊ້ນຳອີກ
- ◆ ເມື່ອມີການນຳເທັກນິກ, ຫຼັກການ ຫຼື ເຄື່ອງມືຊະນິດໃໝ່ເຂົ້າມາປະຍຸກໃຊ້ກໍຈຳເປັນຈະຕ້ອງປັບຂະບວນການໃຫ້ສອດຄ່ອງກັບສິ່ງເລົ່ານັ້ນນຳ
- ◆ ຍຸດທະສາດໃນການປັບປຸງຂະບວນການນັ້ນມີຫລາຍຮູບແບບເຊັ່ນ: Total Quality Management, Business Process Redesign, Continuous Process Improvement, Six Sigma ເປັນຕົ້ນ

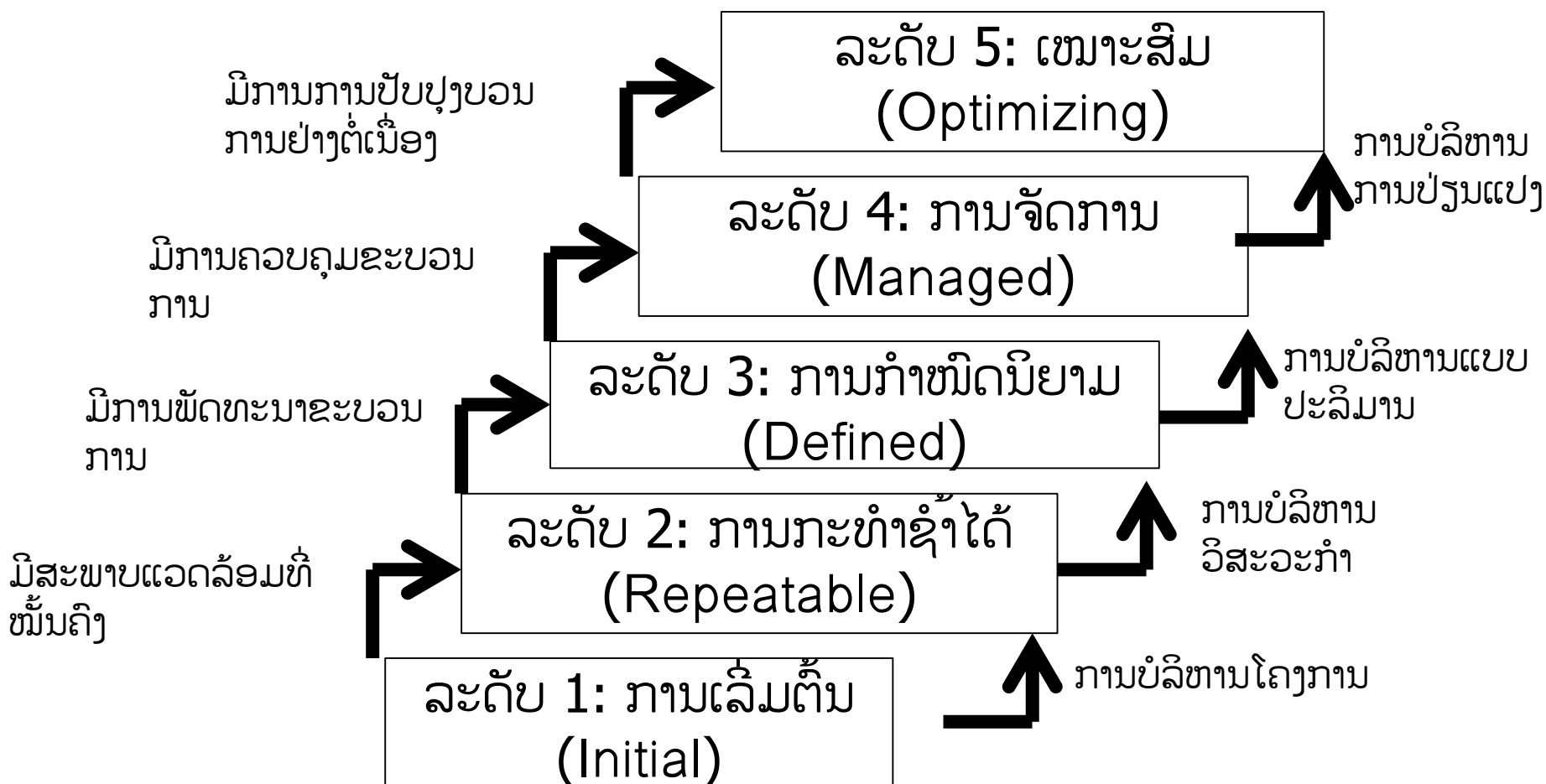


# ການປັບປຸງຂະບວນການດ້ວຍ CMM


- ◆ ແບບຈຳລອງວຸດທິຄວາມສາມາດ (Capability Maturity Model)
  - ◆ ເປັນແບບຈຳລອງທີ່ນຳມາໃຊ້ເພື່ອການປັບປຸງຂະບວນການພັດທະນາຊອບແວຣ໌ໃຫ້ມີຄຸນະພາບ
  - ◆ ແມ່ນແບບຈຳລອງທີ່ມີລັກສະນະເປັນລະດັບຊັ້ນເພື່ອໃຊ້ວັດແທກຄວາມສາມາດຂອງອົງກອນວ່າມີວຸດທິຄວາມສາມາດຢູ່ໃນລະດັບໃດ
  - ◆ ປະກອບດ້ວຍ 5 ລະດັບ: Initial, Repeatable, Defined, Managed, Optimizing
  - ◆ ຈຸດປະສົງແມ່ນເພື່ອຊ່ວຍເຫຼືອອົງກອນທີ່ໜ່ວຍງານຜະລິດຊອບແວຣ໌ໃຫ້ສາມາດປັບປຸງການປະຕິບັດງານຢ່າງເປັນລະບົບ, ມີຄວາມຕໍ່ເນື່ອງແລະ ມີລະບຽບຕາມມາດຕະຖານຊຶ່ງເປັນທີ່ຍອມຮັບ

# ການປັບປຸງຂະບວນການດ້ວຍ CMM

◆ ແບບຈຳລອງວຸດທິຄວາມສາມາດ (Capability Maturity Model)



# ການປັບປຸງຂະບວນການດ້ວຍ CMM



◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 1


- ຄຸນລັກສະນະ

- ບໍ່ເປັນລະບຽບ
- ບໍ່ສາມາດກະທໍາຊ້າ ຫຼື ນໍາມາໃຊ້ຄືນໄດ້
- ມີຄວາມສ່ຽງສູງຫລາຍ

- ຂະບວນການໃນແຕ່ລະດ້ານ

- ບໍ່ມີການກໍາໜົດ

# ການປັບປຸງຂະບວນການດ້ວຍ CMM



◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 2

- ຄຸນລັກສະນະ

- ມີນະໂຍບາຍຈະແຈ້ງ
- ສາມາດກະທໍາຊ້ໍາໄດ້
- ບໍ່ມີການປັບປຸງ

# ການປັບປຸງຂະບວນການດ້ວຍ CMM

◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 2

- ຂະບວນການໃນແຕ່ລະດ້ານ

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

# ການປັບປຸງຂະບວນການດ້ວຍ CMM

◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 3

- ຄຸນລັກສະນະ

- ມີການປັບປຸງປະສິດທິພາບໃນດ້ານຕ່າງໆເຊັ່ນ: ຕົ້ນທຶນ, ກໍານົດເວລາ, ຄຸນນະພາບ, ຄວາມສ່ຽງ

- ຂະບວນການໃນແຕ່ລະດ້ານ


- ການຈັດການຂະບວນການດ້ວຍເອກະສານ

- ການປັບປຸງໃໝ່, ການຝຶກອົບໂຮມ, ການຈັດການບຸກຄົນ

- ການຈັດການຄຸນນະພາບເບື້ອງຕົ້ນ

- ການສະໜັບສະໜູນການຜະລິດຊອບແວຣ໌

# ການປັບປຸງຂະບວນການດ້ວຍ CMM



◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 4

- ຄຸນລັກສະນະ

- ມີປະສິບການ

- ມີການປັບປຸງປະສິທິພາບຢ່າງຕໍ່ເນື່ອງ

- ຂະບວນການໃນແຕ່ລະດ້ານ

- ການຈັດການຂະບວນການທາງດ້ານປະລິມານ

- ການຈັດການຄຸນນະພາບຊອບແວຣ໌

# ການປັບປຸງຂະບວນການດ້ວຍ CMM

◆ ຄຸນລັກສະນະ ແລະ ຂະບວນການສໍາຄັນຂອງວຸດທິຄວາມສາມາດ

◆ ລະດັບ 5

- ຄຸນລັກສະນະ
  - ມີປະສິດທິພາບໃນທຸກໆດ້ານ
  - ມີການປັບປຸງການຮຽນຮູ້
  - ມີການສະສົມປະສົບການ
  - ມີຜູ້ຊ່ຽວຊານ
  - ມີຄຸນນະພາບສູງ, ມີຄວາມສ່ຽງໜ້ອຍ
- ຂະບວນການໃນແຕ່ລະດ້ານ
  - ການສ້າງສັນການປັບປຸງໃໝ່
  - ການບໍລິຫານຄວາມປ່ຽນແປງ
  - ການຈັດສັນສັບພະຍາກອນ



# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Project Management Application

- Microsoft Project

### ◆ Drawing/Graphic Application

- Rational Rose, Visible Analyst, Visual Paradigm, Smart Draw, Visio

### ◆ Word Processor/ Text Editor

### ◆ Integrated Development Environment (IDE)

### ◆ Computer-Aided System Engineering(CASE) Tool

### ◆ Database Management Application

### ◆ Code Generator Tool

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ເປັນຊອບແວທີ່ຊ່ວຍສະໜັບສະໜູນການເຮັດວຽກໃນກິດຈະກຳຕ່າງໆຂອງວຽກງານວິສະວະກຳຊອບແວ
  - ການກຳນົດຄວາມຕ້ອງການ, ການອອກແບບ, ການຂຽນໂປຣແກຣມ, ການທົດສອບໂປຣແກຣມ
- ເປັນເທັກໂນໂລຢີທີ່ເພີ່ມຄວາມສາດໃຫ້ແກ່ຊອບແວ ຊຶ່ງກາຍເປັນເຄື່ອງມືຊ່ວຍແບ່ງເບົາພາລະຂອງນັກພັດທະນາລະບົບ
- ຊະນິດຂອງ CASE Tool
  - CASE Tool ມີຫລາຍປະເພດ ໂດຍມີຫລັກການຈຳແນກປະເພດຫລາຍຢ່າງເຊັ່ນ: ຈຳແນກຕາມໜ້າທີ່, ຈຳແນກຕາມຂະບວນການເຮັດວຽກ, ຈຳແນກຕາມການປະສານງານ

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ຊະນິດຂອງ CASE Tool ຈຳແນກຕາມຂະບວນເຮັດວຽກ
  - ເຄື່ອງມືວິເຄາະຄວາມຕ້ອງການ (Software Requirement Tools)
  - ເຄື່ອງມືອອກແບບຊອບແວ (Software Design Tools)
  - ເຄື່ອງສ້າງຊອບແວ (Software Construction Tools)
  - ເຄື່ອງມືທົດສອບຊອບແວ (Software Testing Tools)
  - ເຄື່ອງມືບຳລຸງຮັກສາຊອບແວ (Software Maintenance Tools)
  - ເຄື່ອງມືຈັດການໂຄງຮ່າງ (Software Configuration Management Tools)
  - ເຄື່ອງມືບໍລິຫານວຽກງານວິສະວະກຳຊອບແວ (Software Engineering Management Tools)
  - ເຄື່ອງມືກວດສອບຄຸນນະພາບ (Software Quality Tools)

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ຊະນິດຂອງ CASE Tool ຈຳແນກຕາມຂະບວນເຮັດວຽກ
  - ເຄື່ອງມືວິເຄາະຄວາມຕ້ອງການ (Software Requirement Tools)  
ແບ່ງເປັນ 2 ກຸ່ມ
    1. ເຄື່ອງມືໃນການການສ້າງແບບຈຳລອງຄວາມຕ້ອງການ(Requirement Modeling Tools)
    2. ເຄື່ອງມືຕິດຕາມຄວາມຕ້ອງການ(Requirement Traceability Tools)
  - ເຄື່ອງມືອອກແບບຊອບແວ (Software Design Tools)
    - ເປັນເຄື່ອງມືທີ່ໃຊ້ສ້າງ ແລະ ກວດສອບການອອກແບບຊອບແວ
    - ປະຈຸບັນມີຢູ່ເປັນຈຳນວນຫຼາຍເຊັ່ນ Rational Rose, EA

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ຊະນິດຂອງ CASE Tool ຈຳແນກຕາມຂະບວນເຮັດວຽກ
  - ເຄື່ອງສ້າງຊອບແວ (Software Construction Tools)
    - ເປັນເຄື່ອງມືທີ່ສະໜັບສະໜູນວຽກໃນການສ້າງຊອບແວທັງໝົດ
    - ໄດ້ແກ່ ເຄື່ອງມືແກ້ໄຂໂປຣແກຣມ(Program Editor), Compiler, Interpreter, Debugger
    - ຕົວຢ່າງເຊັ່ນ Eclipse, EditPlus
  - ເຄື່ອງມືທົດສອບຊອບແວ (Software Testing Tools)
    - ເຄື່ອງສ້າງກໍລະນີທົດສອບ(Testing Generation)
    - ກອບການປະຕິບັດງານທົດສອບ(Test Execution Framework)
    - ເຄື່ອງມືປະເມີນການທົດສອບ(Test Evaluation Tools)
    - ເຄື່ອງມືບໍລິຫານການທົດສອບ(Test Management Tools)
    - ເຄື່ອງມືວິເຄາະປະສິດທິພາບຂອງຊອບແວ(Performance Analysis Tools)

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ຊະນິດຂອງ CASE Tool ຈຳແນກຕາມຂະບວນເຮັດວຽກ
- ເຄື່ອງມືບຳລຸງຮັກສາຊອບແວ (Software Maintenance Tools)  
ເປັນເຄື່ອງມືທີ່ໃຊ້ບຳລຸງຮັກສາຊອບແວ ແບ່ງເປັນ 2 ກຸ່ມ
  - ເຄື່ອງມືສ້າງຄວາມເຂົ້າໃຈ(Comprehension Tools)
  - ເຄື່ອງມືຮີ້ປັບລະບົບໃໝ່(Reengineering Tools)
- ເຄື່ອງມືຈັດການໂຄງຮ່າງ (Software Configuration Management Tools)
  - ເປັນເຄື່ອງມືທີ່ໃຊ້ຕິດຕາມການປ່ຽນແປງຂອງທຸກອົງປະກອບຂອງຊອບແວ ຈັດການຮຸ້ນຂອງຊອບແວ ແລະ ການວາງຈຳໜ່າຍຊອບແວ

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ເຄື່ອງມື (Tools)

### ◆ Computer-Aided System Engineering(CASE) Tool

- ຊະນິດຂອງ CASE Tool ຈຳແນກຕາມຂະບວນເຮັດວຽກ
- ເຄື່ອງມືບໍລິຫານວຽກງານວິສະວະກຳຊອບແວ (Software Engineering Management Tools)
  - ເຄື່ອງມືວາງແຜນຕິດຕາມໂຄງການ(Project Planning and Tracking)
  - ເຄື່ອງມືຈັດການຄວາມສ່ຽງ(Risk Management)
  - ເຄື່ອງມືວັດຜົນໂຄງການ(Measurement)
- ເຄື່ອງມືກວດສອບຄຸນນະພາບ (Software Quality Tools)
  - ເຄື່ອງມືກວດສອບຄຸນນະພາບ(Inspection Tools)
  - ເຄື່ອງມືວິເຄາະຄຸນນະພາບ(Static Analysis Tools)

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ວິທີການ (Methodology)

◆ ເປັນວິທີການໃນການປະຕິບັດງານໃນທາງວິສະວະກຳຊອບແວ ຊຶ່ງໄດ້ກ່າວໃນບົດທີ 1 ແລ້ວ 2 ວິທີຄື: Structure Approach ແລະ Object-Oriented Approach. ນອກຈາກນັ້ນມີວິທີອື່ນໆອີກ ເຊັ່ນ:

- Heuristic Methodology
  - ເປັນວິທີທີ່ບໍ່ເອົາວິທີການທາງຄະນິດສາດເຂົ້າໄປໃຊ້ໃນຂັ້ນຕອນຕ່າງໆ
  - ປະກອບດ້ວຍ: Structure Methodology/Approach, Object-Oriented Methodology, Data-Oriented Methodology
- Formal Methodology
  - ການບອກຂໍ້ກຳໜົດຢ່າງມີແບບແຜນ (Formal Specification)
  - ການກວດສອບຄືນຢ່າງມີແບບແຜນ (Formal Verification)



# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ



## ◆ ວິທີການ (Methodology)

- Heuristic Methodology (Informal Method)
  - ເປັນວິທີທີ່ບໍ່ໄດ້ໃຊ້ຫຼັກການທາງຄະນິດສາດເຂົ້າໃນຂັ້ນຕອນຕ່າງໆ
  - ມີສ່ວນປະກອບດັ່ງນີ້:
    - Structure Approach
    - Object-Oriented Approach
    - Data-Oriented Approach

# ເຄື່ອງມື ແລະ ວິທີການທີ່ໃຊ້ໃນວິສະວະກຳຊອບແວ

## ◆ ວິທີການ (Methodology)

- Formal Methodology

- ເປັນວິທີທີ່ນຳໃຊ້ຫຼັກການທາງຄະນິດສາດເປັນພື້ນຖານ
- ການເຮັດວຽກມີ 2 ຊະນິດ
  - Formal Specification ເປັນວິທີອະທິບາຍຂໍ້ກຳນົດຫຼັກການຄະນິດສາດໃດໜຶ່ງເຊັ່ນ: ພຶດຊະຄະນິດ, ແບບຈຳລອງທາງຄະນິດສາດ
  - Formal Verification ເປັນວິທີການກວດສອບໂດຍໃຊ້ຫຼັກການພິສູດທາງຕົກກະສາດ