252SE311: ວິສະວະກຳຊອບແວຣ໌

ການສ້າງ, ທົດສອບ ແລະ ບຳລຸງຮັກສາ

ບົດທີ 13

ການບຳລຸງຮັກສາຊອບແວຣ໌

Software Maintenance

ເນື້ອໃນຫຍໍ້

- ການບຳລຸງຮັກສາຊອບແວຣ໌
- ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌
- ຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ໌
- ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌
  - ○ເທັກນິກໃນການບຳລຸງຮັກສາຊອບແວຣ໌
  - ○ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

ການບຳລຸງຮັກສາຊອບແວຣ໌

- ແມ່ນການປ່ຽນແປງຊອບແວຣ໌ພາຍຫລັງການສົ່ງມອບ ເພື່ອແກ້ໄຂຂໍ້ຜິດພາດ, ປັບປຸງປະສິດທິພາບ ຫຼື ດັດແປງຊອບແວຣ໌ໃຫ້ເໝາະສົມກັບສະພາບແວດລ້ອມທີ່ປ່ຽນແປງ
- ນອກຈາກນັ້ນ ຍັງເປັນການປັບປຸງແກ້ໄຂໂຄດ ແລະ ເອກະສານ ໃຫ້ຖືກຕ້ອງ ເມື່ອມີຂໍ້ຜິດພາດເກີດຂຶ້ນ ຫຼື ເມື່ອມີການຮ້ອງຂໍໃຫ້ປັບປຸງ

ການບຳລຸງຮັກສາຊອບແວຣ໌

- ຄວາມສຳຄັນຂອງການບຳລຸງຮັກສາ
  - ການປ່ຽນແປງເປັນເຫດໃຫ້ມີການບຳລຸງຮັກສາຊອບແວຣ໌
  - ເນື່ອງຈາກຊອບແວຣ໌ຕ້ອງເຮັດວຽກຢູ່ພາຍໄຕ້ສະພາບແວດລ້ອມຕ່າງໆ ທີ່ມີການປ່ຽນແປງຕະຫລອດເວລາ
  - ເພື່ອໃຫ້ແນ່ໃຈວ່າຊອບແວຣ໌ຍັງຄົງເຮັດວຽກໄດ້ກົງກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້
  - ສິ່ງທີ່ວິສະວະກອນຊອບແວຣ໌ເຮັດໄດ້ໃນລະຫວ່າງການຜະລິດຊອບແວຣ໌ ເພື່ອອຳນວຍຄວາມສະດວກໃຫ້ແກ່ການບຳລຸງຮັກສາ ກໍ່ຄື ການອອກແບບຊອບແວຣ໌ໃຫ້ສາມາດປັບປຸງ ຫຼື ແກ້ໄຂໄດ້ງ່າຍ ແລະ ບໍ່ສິ້ນເປືອງງົບປະມານ

ການບຳລຸງຮັກສາຊອບແວຣ໌

- ການເລືອກລະຫວ່າງບຳລຸງຮັກສາ ແລະ ສ້າງຂຶ້ນໃໝ່
  - ເມື່ອມີການປ່ຽນແປງແກ້ໄຂຊອບແວຣ໌ ໝາຍເຖິງອົງກອນຈະຕ້ອງໄດ້ຈ່າຍເງິນຄ່າປັບປຸງ ສະນັ້ນ ອົງກອນຈະຕ້ອງໄດ້ພິຈາລະນາ ຕັດສິນໃຈເລືອກລະຫວ່າງການບຳລຸງຮັກສາ ແລະ ການພັດທະນາຂຶ້ນໃໝ່ ວ່າອັນໃດຄຸ້ມຄ່າກ່ວາກັນ
  - ສາມາດໃຊ້ຄຳຖາມເລົ່ານີ້ເປັນແນວທາງໃນການຕັດສິນໃສ
    - ຕົ້ນທຶນການບຳລຸງຮັກສາຫລາຍເກີນໄປບໍ?
    - ຊອບແວຣ໌ຫຼືລະບົບນັ້ນບໍ່ມີຄວາມໜ້າເຊື່ອຖືອີກແລ້ວບໍ?
    - ຊອບແວຣ໌ນັ້ນບໍ່ສາມາດດັດແປງແກ້ໄຂໄດ້ອີກແລ້ວບໍ?
    - ປະສິດທິພາບຂອງຊອບແວຣ໌ຍັງມີພຽງພໍທີ່ຈະເຮັດວຽກພາຍໄຕ້ເງື່ອນໄຂຕ່າງບໍ່?
    - ມີຊອບແວຣ໌ອື່ນທີ່ສາມາດເຮັດວຽກໄດ້ດີກ່ວາບໍ່?

    ຖ້າຄຳຕອບສ່ວນຫຼາຍວ່າ ແມ່ນແລ້ວ ໝາຍເຖິງວ່າຈະຕ້ອງໄດ້ປ່ຽນຊອບແວຣ໌ໃໝ່

ການບຳລຸງຮັກສາຊອບແວຣ໌

- ປະເພດຂອງການບຳລຸງຮັກສາ
  - ການບຳລຸງຮັກສາມີລັກສະນະສຳຄັນ 4 ປະການດັ່ງນີ້
    - ຄວບຄຸມການເຮັດວຽກຂອງແຕ່ລະ Function ຂອງຊອບແວຣ໌ໃຫ້ຖືກຕ້ອງເປັນປະຈຳທຸກວັນ
    - ຄວບຄຸມການປັບປຸງຊອບແວຣ໌
    - ຮັກສາສະພາບຄວາມສົມບຸນຂອງ function ຂອງຊອບແວຣ໌ໄວ້ສະເໝີ
    - ປ້ອງກັນບໍ່ໃຫ້ປະສິດທິພາບຂອງຊອບແວຣ໌ລຸດລົງເກີນລະດັບທີ່ສົມຄວນ
  - ຈາກລັກສະນະສຳຄັນ 4 ຢ່າງໄດ້ແບ່ງປະເພດການບຳລຸງຮັກສາອອກເປັນ 4 ປະເພດ
    - Corrective Manintenance ເປັນການແກ້ໄຂຂໍ້ຜິດພາດທັນທີທີ່ພົບ
    - Adaptive Maintenance ເປັນການດັດແປງສ່ວນທີ່ໄດ້ຮັບຜົນກະທົບ
    - Perfective Maintenance ເປັນການປັບປຸງປະສິດທິພາບການເຮັດວຽກ
    - Preventive Maintenance ເປັນການປ່ຽນແປງເພື່ອປ້ອງກັນບໍ່ໃຫ້ເຮັດວຽກຜິດພາດ

ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌

- ການບຳລຸງຮັກສາບໍ່ໃຫ້ກະທົບຕໍ່ທຸລະກິດຂອງລູກຄ້າ
  - ການບຳລຸງຮັກສາຊອບແວຣ໌ເປັນເລື່ອງຍຸ້ງຍາກ
  - ຕ້ອງເຮັດໃນຄະນະທີ່ຜູ້ໃຊ້ກຳລັງໃຊ້ງານຢູ່
  - ອາດເຮັດໃຫ້ເກີດຄວາມເສຍຫາຍຕໍ່ທຸລະກິດຂອງຜູ້ໃຊ້
  - ບໍ່ເຮັດໃຫ້ການເຮັດວຽກຂອງຜູ້ໃຊ້ຢຸດສະງັກ

ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌

- ບັນຫາດ້ານທີມງານ
  - ຂໍ້ຈຳກັດດ້ານການສ້າງຄວາມເຂົ້າໃຈ
    - ຍາກໃນການສ້າງຄວາມເຂົ້າໃຈກັບເອກະສານ ແລະ ໂຄດທີ່ມີຢູ່ຈຳນວນຫລວງຫລາຍ
    - ຂາດທັກສະໃນການສ້າງຄວາມເຂົ້າໃຈໃນສິ່ງທີ່ຕົນເອັງຮັບຜິດຊອບ
  - ກຳລັງໃຈ
    - ທີມງານບຳລຸງຮັກສາສ່ວນໃຫຍ່ຈະຮູ້ສຶກວ່າຕົນເອັງເປັນທີມງານທີ່ບໍ່ມີຄວາມສຳຄັນເທົ່າກັບທີມງານຂຽນໂປຣແກຣມ
  - ການຕັດສິນໃຈດ້ານການບໍລິຫານເປັນເລື່ອງສຳຄັນ
    - ທີມງານຈະຕ້ອງໄດ້ປະຕິບັດຕາມການຕັດສິນໃຈຂອງຜູ້ບໍລິຫານ

ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌

- ບັນຫາດ້ານເທັກນິກ
  - ບາງລະບົບຖືກພັດທະນາຂຶ້ນມາໃຊ້ເປັນໄລຍະເວລາດົນແລ້ວ ຊຶ່ງມີບາງຢ່າງທີ່ທີມງານປະຈຸບັນບໍ່ເຄີຍຮູ້ມາກອນ ເຊັ່ນ ດ້ານສະຖາປັດຕະຍະກຳ ແລະ ຂັ້ນຕອນການຜະລິດທີ່ລະບົບເກົ່າໃຊ້ ເຮັດໃຫ້ການກວດສອບເພື່ອບຳລຸງຮັກສາຍາກ
  - ສະຖາປັດຕະຍະກຳ ແລະ ວິທີທາງແບບເກົ່າ
    - ໃຊ້ສະຖາປັດຕະຍະກຳແບບເດີມ ແລະ ວິທີທາງການຂຽນໂປຣແກຣມແບບເດີມ ຊຶ່ງຂ້ອນຂ້າງຊັບຊ້ອນແລະບໍ່ຢືດຢຸ່ນ ເຮັດໃຫ້ການປະເມີນ ແລະ ສ້າງຄວາມເຂົ້າໃຈຍາກ
  - ຄວາມຍາກຂອງການທົດສອບ
    - ການຊອກຫາເວລາໃນການທົດສອບຍາກ (ລະບົບບໍ່ຫວ່າງເພື່ອໃຫ້ທົດສອບ)

ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌

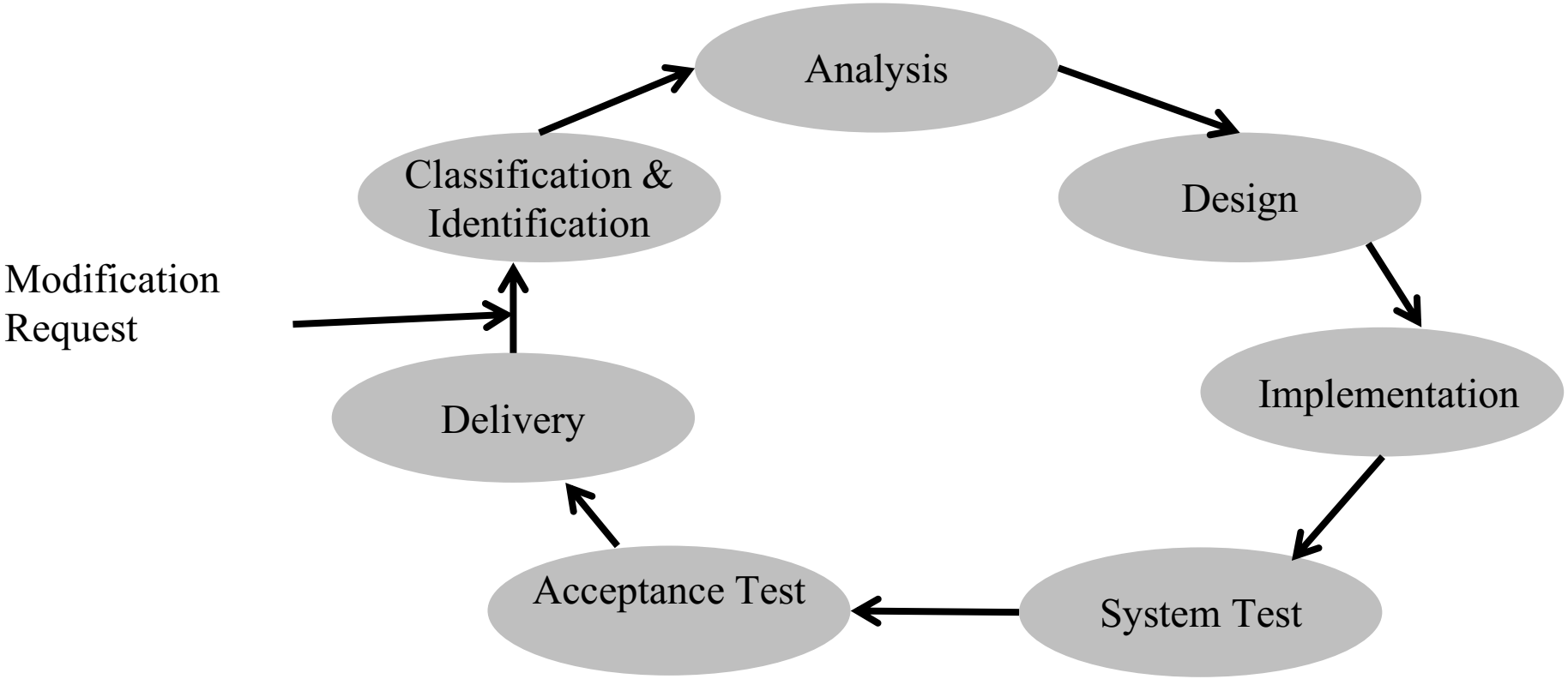
- ບັນຫາຈາກການປາມີປາມອມ
  - ສິ່ງທີ່ຜູ້ໃຊ້ສະເໜີໃຫ້ປ່ຽນແປງແກ້ໄຂອາດເປັນເລື່ອງຍາກສຳຫລັບທີມງານບຳລຸງຮັກສາ ສະນັ້ນ ຜູ້ຮັບຜິດຊອບຕ້ອງເຮັດໜ້າທີ່ໄກ່ເກ່ຍໃຫ້ປາມີປາມອມໃນຂໍ້ຄັດແຍ່ງດັ່ງກ່າວ
  - ເມື່ອມີການປາມີປາມອມເກີດຂຶ້ນ ບັນຫາຈຶ່ງຂຶ້ນຢູ່ກັບທີມງານບຳລຸງຮັກສາເຊັ່ນ:
    - ອາດຕ້ອງມີການຈ້າງທີມງານອື່ນເພີ່ມເພື່ອໃຫ້ແລ້ວໄວ,
    - ທີມງານໃໝ່ທີ່ຍັງບໍ່ຄຸ້ນເຄີຍກັບລະບົບຫຼືຊອບແວຣ໌ຈຶ່ງເຮັດໃຫ້ຍາກຂຶ້ນຕື່ມ
    - ການເຮັດວຽກເພື່ອໃຫ້ແລ້ວໄວເຮັດໃຫ້ບໍ່ມີປະສິດທິພາບ

ບັນຫາຂອງການບຳລຸງຮັກສາຊອບແວຣ໌

- ບັນຫາດ້ານຕົ້ນທຶນ
  - ຕົ້ນທຶນໃນການບຳລຸງຮັກສາຂ້ອນຂ້າງສຸງ ເນື່ອງຈາກຕ້ອງການແຮງງານ ແລະ ເວລາ ໂດຍມີສາເຫດດັ່ງນີ:
    - ປະເພດຂອງຊອບແວຣ໌ ຊອບແວຣ໌ບາງຢ່າງບຳລຸງຮັກສາຍາກ
    - ຊອບແວຣ໌ແບບແປກໃໝ່
    - ການຍົກຍ້າຍ ແລະ ຈຳນວນແຮງງານທີ່ມີຢູ່
    - ຊ່ວງຊີວິດຂອງຊອບແວຣ໌
    - ສະພາບແວດລ້ອມທີ່ປ່ຽນແປງສະເໝີ
    - ຄຸນນະພາບຂອງຮາດແວຣ໌, ການອອກແບບ, ໂຄດ, ເອກະສານ ແລະ ການທົດສອບ

ຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ์

↳ ແຜນວາດສະແດງຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ์



ຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ການສະເໜີປ່ຽນແປງແກ້ໄຂ (Modification Request)
  - ການສະເໜີຫຼືຮອ້ງຂໍໃຫ້ປ່ຽນແປງແກ້ໄຂຊອບແວຣ໌ຈາກຜູ້ໃຊ້
- ການຈຳແນກ ແລະ ກຳໜົດປະເພດການບຳລຸງຮັກສາ
  - ກຳໜົດໝາຍເລກຫຼືລະຫັດ ພ້ອມທັງຈຳແນກປະເພດຂອງການບຳລຸງຮັກສາ
- ການວິເຄາະ (Analysis)
  - ວິເຄາະຄວາມເປັນໄປໄດ້ຂອງການປ່ຽນແປງຕາມຂໍ້ສະເໜີ
  - ວິເຄາະລາຍລະອຽດເພື່ອກຳໜົດໜ້າວຽກຕ່າງໆເຊັ່ນ: ກຳໜົດຂອບເຂດ, ກຳໜົດເທັກນິກ, ເທັກນິກໃນການທົດສອບ ແລະ ວາງແຜນດຳເນີນງານ ເປັນຕົ້ນ

ຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ການອອກແບບ (Design)
  - ອອກແບບໂມດູ່ນທີ່ຕ້ອງໄດ້ປ່ຽນແປງແກ້ໄຂ ແລະ ໂມດູນອື່ນທີ່ໄດ້ຮັບຜົນກະທົບທັ້ງໝົດ
  - ແກ້ໄຂເອກະສານທັ້ງໝົດທີ່ກ່ຽວຂ້ອງ
  - ອອກແບບກໍລະນີທົດສອບສຳຫລັບໂມດູ່ນໃໝ່ທີ່ຜ່ານການແກ້ໄຂ
  - ພິຈາລະນາເອກະສານຂໍ້ກຳໜົດຄວາມຕ້ອງການເພື່ອປັບປຸງໃຫ້ຖືກລຸ້ນຂອງຊອບແວຣ໌
- ການດຳເນີນການບຳລຸງຮັກສາ (Implementation)
  - ປ່ຽນແປງແກ້ໄຂໂຄດໃນສ່ວນທີ່ຕ້ອງການປ່ຽນແປງເທື່ອລະອັນ ແລ້ວເອົາມາລວມເຂົ້າກັນເພື່ອທົດສອບຕາມເທັກນິກທີ່ໄດ້ກຳໜົດໄວ້ໃນຂັ້ນຕອນການວິເຄາະ
  - ວິເຄາະຄວາມສ່ຽງ

ຂັ້ນຕອນໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ການທົດສອບລະບົບ (System Test)
  - ທົດສອບຊອບແວຣ໌ໂດຍເລີ່ມຕົ້ນຈາກລະດັບຫົວໜ່ວຍຍ່ອຍ, ລະດັບລວມ, ຈົນຮອດທົດສອບລະບົບ ເພື່ອຮັບປະກັນວ່າມັນເຮັດວຽກໄດ້ຖືກຕ້ອງ
- ການທົດສອບການຍອມຮັບ (Acceptance Test)
  - ເປັນການທົດສອບເພື່ອເຮັດໃຫ້ແນ່ໃນວ່າລະບົບ ແລະ ຊອບແວຣ໌ລຸ້ນໃໝ່ທີ່ໄດ້ຮັບການປັບປຸງແລ້ວນັ້ນເປັນທີ່ຍອມຮັບຂອງຜູ້ໃຊ້
- ການສົ່ງມອບ (Delivery)
  - ການວາງແຜນສົ່ງມອບ, ບອກໃຫ້ຜູ້ໃຊ້ຮູ້ຈັກວິທີການຕິດຕັ້ງ, ຝຶກອົບຮົມ, ເກັບສຳຮອງຊອບແວຣ໌ລຸ້ນເກົ່າແລະໃໝ່ໄວ້, ຕຽມລະບົບສະໜັບສະໜຸນການໃຊ້ງານຂອງຜູ້ໃຊ້

ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ເທັກນິກໃນການບຳລຸງຮັກສາຊອບແວຣ໌
  - ການສ້າງຄວາມເຂົ້າໃຈໂປຣແກຣມ
    - ອ່ານ ແລະ ສ້າງຄວາມເຂົ້າໃຈໃນໂປຣແກຣມເພື່ອປ່ຽນແປງໂຄດ ໂດຍອາໃສເຄື່ອງມື Code Browser ຊຶ່ງມີກົນໄກໃນການວິເຄາະໂຄດໄດ້ຫລາຍຢ່າງ
  - ການຟື້ນຟູສະພາບຂອງຊອບແວຣ໌
    - ການປັບປ່ຽນເອກະສານໃໝ່ (Redocumentation)
    - ການປັບປ່ຽນໂຄງສ້າງໃໝ່ (Restructuring)
    - ການເຮັດວິສະວະກຳຍ້ອນກັບ (Reverse Engineering)
    - ການປັບປ່ຽນວິສະວະກຳໃໝ່ (Reengineering)

ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ການປັບປ່ຽນເອກະສານໃໝ່ (Redocumentation)
  - ເປັນການສ້າງເອກະສານຂອງລະບົບຂຶ້ນມາໃໝ່ ໂດຍອາໃສການວິເຄາະຈາກ ຊອດໂຄດຂອງຊອບແວຣ໌ລະບົບເກົ່າເຊັ່ນ:
    - ການເອີ້ນໃຊ້ຕົວປ່ຽນ
    - ການເອີ້ນໃຊ້ຄອມໂພເນັ້ນ
    - ເສັ້ນທາງການຄວບຄຸມການເຮັດວຽກຂອງໂປຣແກຣມ
    - ການຮັບ-ສົ່ງພາຣາມິເຕີ
    - ເສັ້ນທາງການທົດສອບໂປຣແກຣມ

ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ການປັບປ່ຽນໂຄງສ້າງໃໝ່ (Restructuring)
  - ເປັນການປັບໂຄງສ້າງຂອງຊອດໂຄດເດີມໃຫ້ສາມາດເຂົ້າໃຈແລະແກ້ໄຂໄດ້ງ່າຍຂຶ້ນ
- ການເຮັດວິສະວະກຳຍ້ອນກັບ (Reverse Engineering)
  - ເປັນການເອົາຊອດໂຄດເກົ່າມາວິເຄາະແລ້ວອອກແບບໃໝ່ໃຫ້ມີປະສິດທິພາບ ແລະ ບຳລຸງຮັກສາໄດ້ງ່າຍຂຶ້ນ
- ການປັບປ່ຽນວິສະວະກຳໃໝ່ (Reengineering)
  - ແມ່ນການພິຈາລະນາ ແລະ ປ່ຽນແປງຊອບແວຣ໌ເດີມແລ້ວສ້າງໃຫ້ເປັນຊອບແວຣ໌ໃນຮູບແບບໃໝ່ ຫຼື ເວົ້າວ່າເປັນເປັນການດັດແປງຊອບແວຣ໌ເດີມໃຫ້ເປັນຮູບແບບໃໝ່ ເພື່ອເພີ່ມ Function ໃໝ່ ຫຼື ແກ້ໄຂຂໍ້ບົກພ່ອງຂອງຊອບແວຣ໌ເດີມ

ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌
  - Text Editor
    - ເພື່ອສະແດງໂຄດໂປຣແກຣມທີ່ຕ້ອງການວິເຄາະຫຼືແກ້ໄຂ
    - ສາມາດ copy ໂຄດ ຫຼື ເອກະສານໂປຣແກຣມໄດ້
    - ສາມາດຕິດຕາມການປ່ຽນແປງທີ່ເກີດຂຶ້ນກັບໂຄດຈາກຈຸດໜຶ່ງໄປຫາອີກຈຸດໜຶ່ງໄດ້
  - File Comparator
    - ເປັນຊອບແວຣ໌ທີ່ໃຊ້ປຽບທຽບໂປຣແກຣມຕ່າງເພື່ອເບິ່ງວ່າ File ເຫລົ່ານັ້ນຊ້ຳກັນບໍ່
  - Compiler and Linkers
    - ເປັນເຄື່ອງມືທີ່ໃຊ້ໃນການແປໂຄດໂປຣແກຣມ
  - Debugging Tools
    - ເປັນເຄື່ອງມືທີ່ຊ່ວຍຊອກຫາຂໍ້ຜິດພາດ ຫຼື ຂໍ້ບົກພ່ອງທາງດ້ານຕັກກະຂອງໂປຣແກຣມຢ່າງເປັນລຳດັບຂອງການປະມວນຜົນ

ເທັກນິກ ແລະ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

↳ ເຄື່ອງມືໃນການບຳລຸງຮັກສາຊອບແວຣ໌

- Code Reference Generators
  - ເປັນເຄື່ອງມືທີ່ສ້າງການເຊື່ອມຍົງຂອງອົງປະກອບຕ່າງໆຂອງໂປຣແກຣມ ເຊັ່ນ: ການອ້າງອິງເຖິງຕົວປ່ຽນ, ຄລາດ, ຫຼື ຄອມໂພເນັ້ນ ເພື່ອໃຫ້ທີມງານສາມາດພິຈາລະນາການເຮັດວຽກຮ່ວມກັນລະຫວ່າງອົງປະກອບດັ່ງກ່າວໄດ້ງ່າຍຂຶ້ນ
- Static Code Analyzers
  - ເປັນເຄື່ອງມືວິເຄາະໂຄດແບບບໍ່ປ່ຽນແປງເຊັ່ນ: ວິເຄາະຈຳນວນຊັ້ນຂອງເງື່ອນໄຂ, ຈຳເສັ້ນທາງການເຮັດວຽກ, ຈຳນວນຮອບການວົນລູບ, ຈຳນວນບັນແຖວ ເປັນຕົ້ນ
- Configuration management Repositories
  - ເປັນຖານຂໍ້ມູນສຳຫລັບການຈັດເກັບການປ່ຽນແປງຂອງຊອບແວຣ໌ ແລະ ການປ່ຽນແປງອົງປະກອບອື່ນຂອງລະບົບ ແລະ ສ້າງລາຍງານ