252SE311: ວິສະວະກຳຊອບແວຣ໌

ການສ້າງ, ທົດສອບ ແລະ ບຳລຸງຮັກສາ

ບົດທີ 12

ການທົດສອບຊອບແວຣ໌

Software Testing

ເນື້ອໃນຫຍໍ້

- ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌
- ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ
- ການທົດສອບລວມ
- ການທົດສອບລະບົບ
- ການທົດສອບຊອບແວຣ໌ແບບວັດຖຸ
- ກໍລະນີ ແລະ ການວາງແຜນການທົດສອບ
- ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

- ເປັນກິດຈະກຳທີ່ເຮັດຂຶ້ນເພື່ອປະເມີນ ແລະ ປັບປຸງຄຸນນະພາບຂອງຊອບແວຣ໌ ໂດຍການກວດຫາຂໍ້ຜິດພາດ ແລະ ບັນຫາທີ່ເກີດຂຶ້ນ ແລ້ວແກ້ໄຂຂໍ້ຜິດພາດ ແລະ ບັນຫາໃຫ້ຖືກຕ້ອງ
- ຈຸດປະສົງຂອງການທົດສອບຊອບແວຣ໌ແມ່ນເພື່ອພິສູດວ່າຊອບແວຣ໌ເຮັດວຽກໄດ້ຄົບທຸກໜ້າທີ່ຕາມຂໍ້ກຳໜົດຄວາມຕ້ອງການ ແລະ ແຕ່ລະໜ້າທີ່ສາມາດປະມວນຜົນຂໍ້ມູນໄດ້ຢ່າງຖືກຕ້ອງ

ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

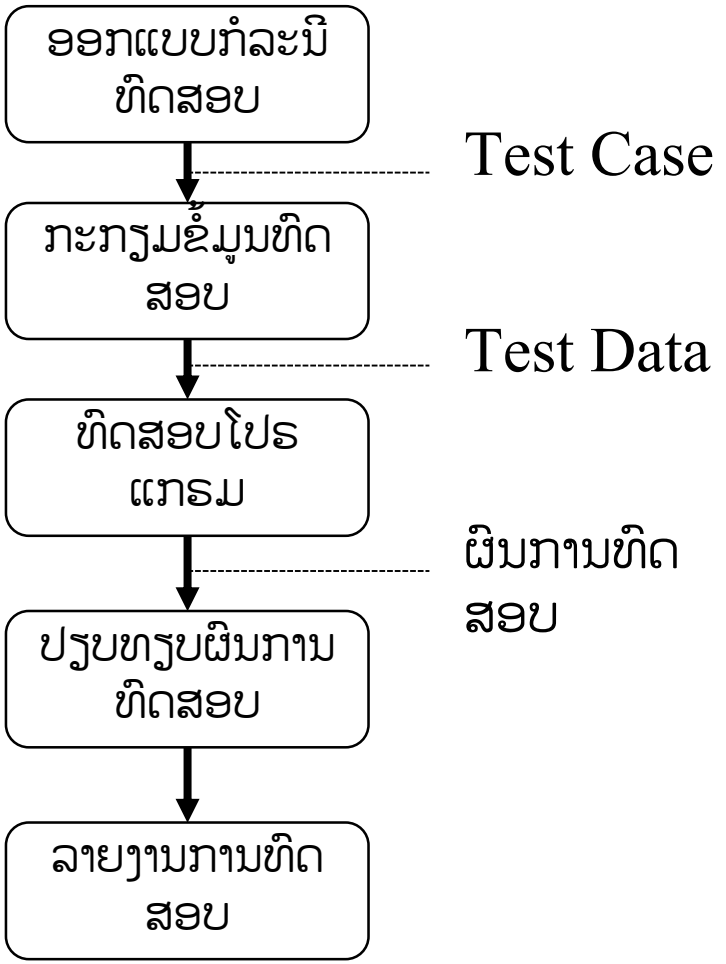
- ຄຳສັບທີ່ຄວນຮູ້ຈັກ
  - Error ແມ່ນການກະທຳຜິດ ໝາຍເຖິງຄ່າຈິງທີ່ໄດ້ຈາກການເຮັດວຽກບໍ່ກົງກັບຄ່າຖືກຕ້ອງ ນອກຈາກນັ້ນ ຍັງລວມເຖິງຜົນການຕັດສິນໃຈຜິດຈາກຄວາມຕ້ອງການ
  - Fault “ຄວາມຜິດພາດ ຫຼື ຂໍ້ບົກພ່ອງ” ໝາຍເຖິງສະພາບທີ່ຂະບວນການປະມວນຜົນຂອງຊອບແວຣ໌ບໍ່ປົກກະຕິ
  - Failure ໝາຍເຖິງຊອບແວຣ໌ຫຼືຮາດແວຣ໌ບໍ່ສາມາດເຮັດວຽກຕາມໜ້າທີ່ໃດໜຶ່ງ ລວມເຖິງບໍ່ສາມາດແຈ້ງເຕືອນຂໍ້ຜິດພາດ

ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

- ລະດັບການທົດສອບຊອບແວຣ໌
  - ການທົດສອບໃນລະດັບຫົວໜ່ວຍຍ່ອຍ (Unit Testing)
    - ເປັນການທົດສອບແຕ່ລະພາກສ່ວນຍ່ອຍສຸດຂອງຊອບແວຣ໌ ເພື່ອປະເມີນການເຮັດວຽກໃນດ້ານຕ່າງໆ
  - ການທົດສອບໃນລະດັບລວມ (Integration Testing)
    - ເປັນການທົດສອບການເຮັດວຽກຂອງກຸ່ມໂປຣແກຣມ ຊຶ່ງເປັນການທົດສອບຫລັງຈາກທີ່ເອົາແຕ່ລະພາກສ່ວນຍ່ອຍມາລວມເຂົ້າກັນ
  - ການທົດສອບລະບົບ (System Testing)
    - ເປັນການທົດສອບຊອບແວຣ໌ເມື່ອເອົາມາລວມເຂົ້າກັບອົງປະກອບອື່ນຂອງລະບົບ

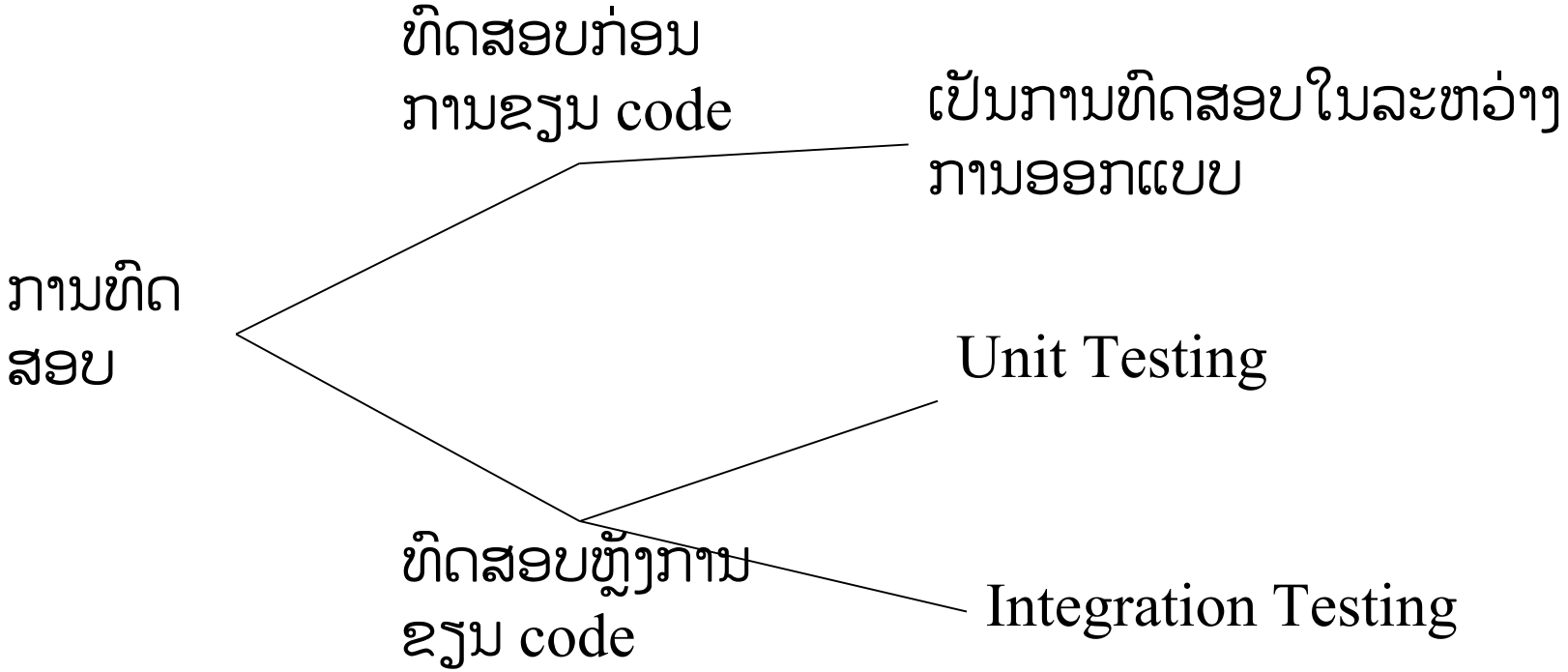
ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

- ວິທີທາງໃນການທົດສອບຊອບແວຣ໌
  - ວິທີທາງໃນການທົດສອບຊອບແວຣ໌ທີ່ດີທີ່ສຸດແມ່ນການທົດສອບຕາມຮອບຂອງການສ້າງຊອບແວຣ໌ ໂດຍເລີ່ມຕົ້ນຈາກການທົດສອບເທື່ອລະໂມດູນ ເອີ້ນວິທີການນີ້ວ່າ Incremental Testing Approach



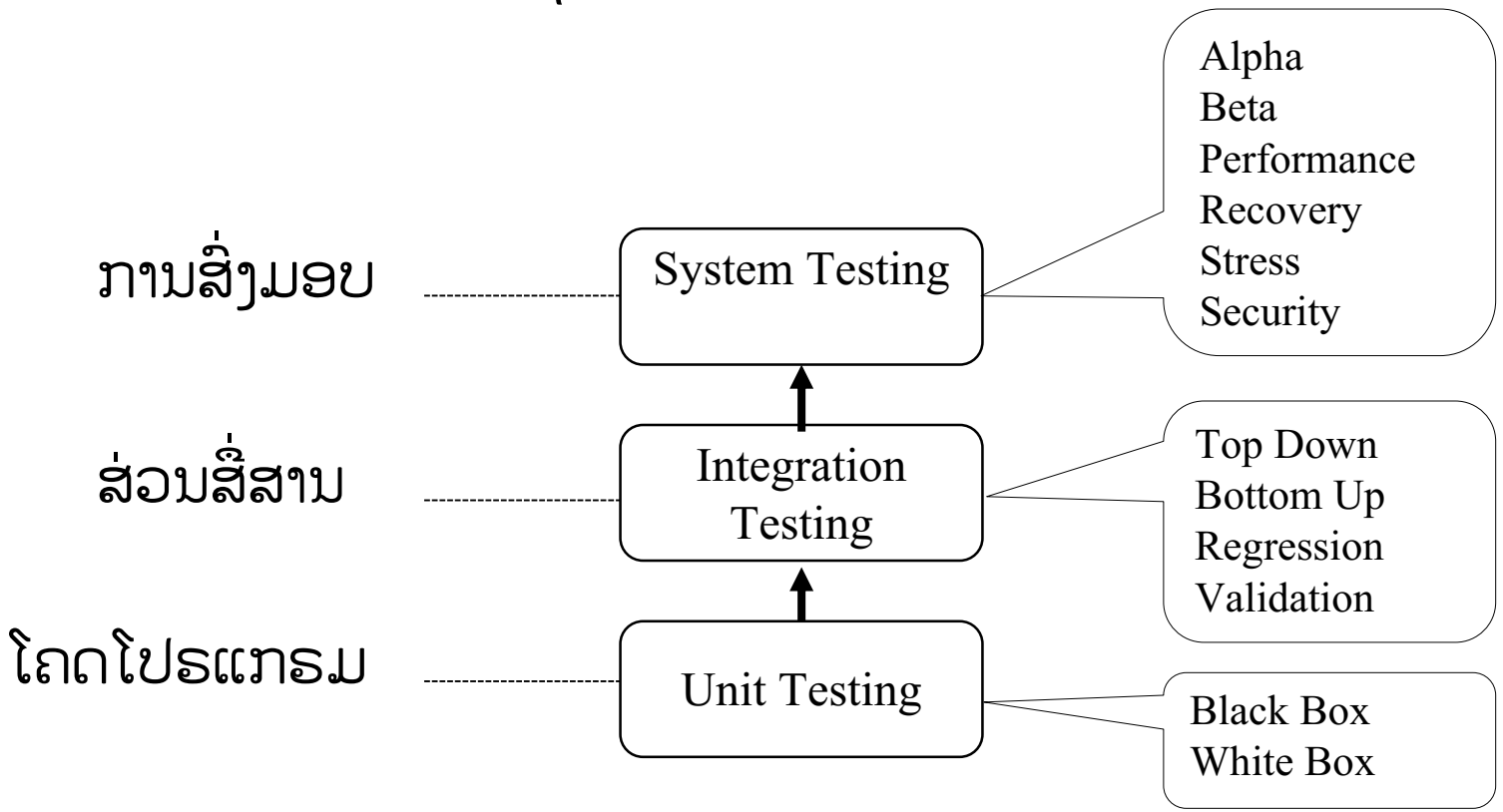
ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

↳ ໂຄງສ້າງຕົ້ນໄມ້ຂອງການທົດສອບຊອບແວຣ໌



ຄວາມຮູ້ເບື້ອງຕົ້ນຂອງການທົດສອບຊອບແວຣ໌

↳ ວິທີການທົດສອບຊອບແວຣ໌



ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ສາມາດທົດສອບຫລາຍໜ່ວຍຍ່ອຍ (Unit) ພ້ອມໆກັນໄດ້
- ສິ່ງທີ່ຕ້ອງທົດສອບໃນລະດັບນີ້ມີດັ່ງນີ້
  - ໂຄງສ້າງຂໍ້ມູນ (Data Structure) - ໂຄງສ້າງຂໍ້ມູນ ພາຍໃນໂມດຸນ
  - ເງື່ອນໄຂຂອງຂອບເຂດ (Boundary Condition) - ຂອບເຂດຄ່າຂໍ້ມູນທີ່ໂປຣແກຣມຕ້ອງປະມວນຜົນ
  - ເສັ້ນທາງການປະມວນຜົນອິດສະຫລະ (Independent Process Path)- ແມ່ນເສັ້ນທາງການເຮັດວຽກທີ່ແຕກຕ່າງກັນຕາມເງື່ອນໄຂ
  - ເສັ້ນທາງການປະມວນຜົນຂໍ້ຜິດພາດ ແລະ ການສະແດງຂໍ້ຜິດພາດ (Error Proccessing Path)

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - White Box Testing
    - ເປັນວິທີທີ່ໃຊ້ເສັ້ນທາງຄວບຄຸມການເຮັດວຽກ ແລະ ໂຄງສ້າງຄວບຄຸມທີ່ໄດ້ຈາກການອອກແບບມາຊ່ວຍອອກແບບກໍລະນີທົດສອບ
    - ທົດສອບສິ່ງຕ່າງໆຕໍ່ໄປນີ້
      1. ທົດສອບທຸກເສັ້ນທາງໃນຂະບວນການຈະຕ້ອງເຮັດວຽກຢ່າງຖືກຕ້ອງ
      2. ທົດສອບການຕັດສິນໃຈທາງຕັກກະສາດທຸກການຕັດສິນໃຈ ທັງຄ່າທີ່ເປັນຈິງ ແລະ ຄ່າທີ່ບໍ່ເປັນຈິງ
      3. ທົດສອບການເຮັດວຽກພາຍໃນລູບຕາມຈຳນວນຄັ້ງຂອງການວົນລູບ
      4. ທົດສອບໂຄງສ້າງຂໍ້ມູນພາຍໃນໃຫ້ຖືກຕ້ອງກ່ອນທີ່ຈະສົ່ງໄປປະມວນຜົນໃນໂປຣແກຣມຫຼືໜ່ວຍອື່ນ

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:

  - White Box Testing
    - ການສ້າງກໍລະນີທົດສອບນັ້ນຈະຕ້ອງໄດ້ສ້າງ Flow chart ຫຼື Graph ສະແດງຂະບວນການເຮັດວຽກ (Process Flow Graph) ຈາກປະໂຫຍກຄຳສັ່ງໃນໂປຣແກຣມ ເພື່ອກຳໜົດເສັ້ນທາງການທົດສອບຈາກ Graph
    - ທົດສອບທຸກໆເສັ້ນທາງ ລວມທັງຄ່າທີ່ເປັນຈິງ ແລະ ຄ່າທີ່ບໍ່ເປັນຈິງ
    - ເອົາຜົນໄດ້ຮັບມາປຽບທຽບກັບຂໍ້ມູນຊຸດທົດສອບ ວ່າເປັນໄປຕາມຜົນຂອງຊຸດທົດສອບບໍ່
    - ຜົນດີ: ສາມາດກວດສອບຫາຂໍ້ຜິດພາດຂອງໂປຣແກຣມໄດ້ຢ່າງລະອຽດ
    - ຜົນເສຍ: ໃຊ້ເວລາຫລາຍ, ຕົ້ນທຶນຫລາຍ

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

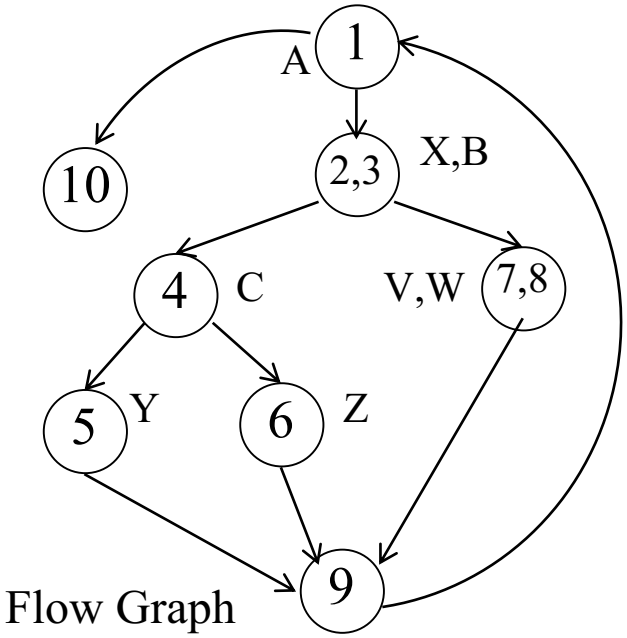
- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - White Box Testing
    - ສົມມຸດວ່າມີໂປຣແກຣມທີ່ຕ້ອງການທົດສອບດັ່ງນີ້

```
function f1()
{
1  while A {
     2  process X;
     3  if B
          4  if C
               5  process Y;
          6  else   process Z;
          else {
          7  process V;
          8  process W;
          }
   9  } //end while
10 } //end function
```

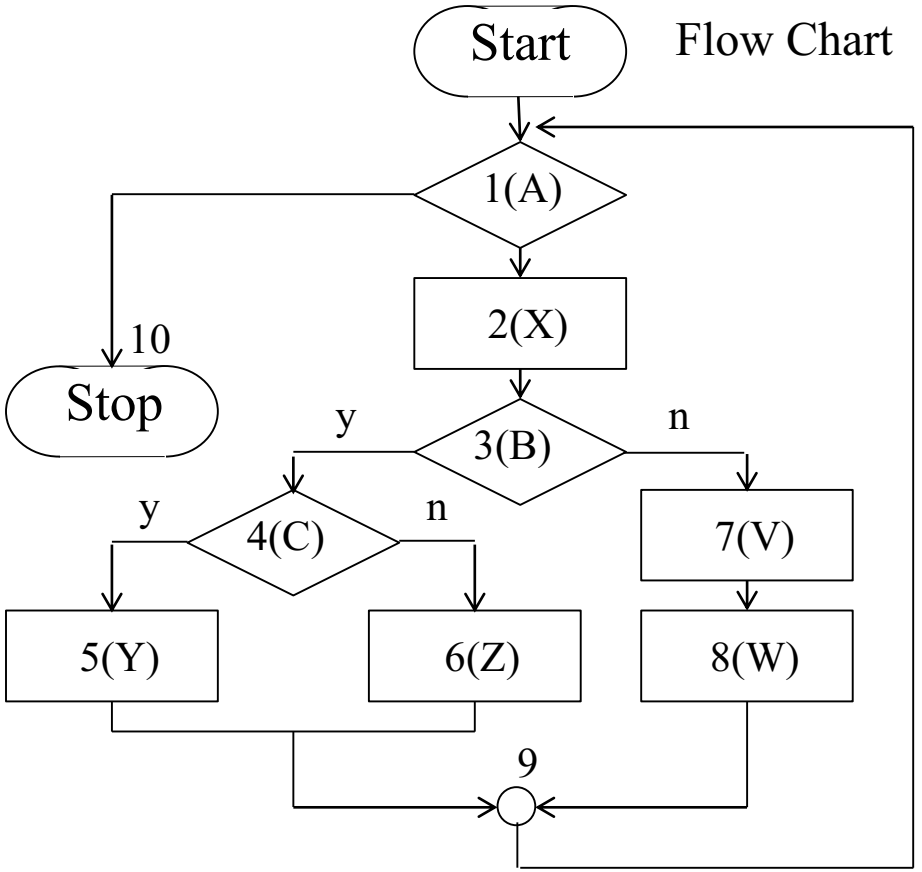
ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - White Box Testing

    ກຳໜົດເສັ້ນທາງການທົດສອບໂປຣແກຣມ



Flow Graph



Flow Chart

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - ຈາກເສັ້ນທາງການເຮັດວຽກຂອງໂປຣແກຣມທັງໝົດເຫັນວ່າມີທັງໝົດ 4 ເສັ້ນທາງທີ່ຕ້ອງທົດສອບ
    1) 1, 10
    2) 1, 2, 3, 4, 6, 9, 1, 10
    3) 1, 2, 3, 4, 5, 9, 1, 10
    4) 1, 2, 3, 7, 8, 9, 1, 10

ການທົດສອບລະດັບຫ້ອໜ່ອຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - Black Box Testing
    - ບາງຄັ້ງເອີ້ນວ່າ Behavioral Testing
    - ແມ່ນການທົດສອບການເຮັດວຽກຂອງຊອບແວຣ໌ໃນແຕ່ລະໜ້າທີ່ຕາມຂໍ້ກຳໜົດຄວາມຕ້ອງການ ເພື່ອເບິ່ງວ່າຊອບແວຣ໌ເຮັດວຽກໄດ້ຖືກຕ້ອງຕາມທີ່ກຳໜົດໄວ້ບໍ່ ໂດຍບໍ່ຄຳນຶງເຖິງຄຳສັ່ງພາຍໃນ
    - ນອກຈາກນັ້ນຍັງເປັນການທົດສອບປະສິດທິພາບຂອງຊອບແວຣ໌ ແລະ ເງື່ອນໄຂຂອບເຂດຂອງຂໍ້ມູນທີ່ປ້ອນເຂົ້າ

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - Black Box Testing
    - ການທົດສອບແບບກ່ອງດຳສາມາດຊ່ວຍໃຫ້ຄົ້ນພົບຂໍ້ຜິດພາດດັ່ງນີ້
      1. ໜ້າທີ່ການເຮັດວຽກຜິດພາດ
      2. ການເຮັດວຽກບໍ່ຄົບໜ້າທີ່
      3. ຄວາມຜິດພາດຂອງພາກສ່ວນສື່ສານກັບລະບົບອື່ນ
      4. ຄວາມຜິດພາດຂອງການຕັດສິນໃຈເຮັດວຽກຕໍ່ຫຼືຢຸດເຮັດວຽກ
      5. ຄວາມຜິດພາດຂອງການປະມວນຜົນຂໍ້ມູນ

ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ

- ວິທີການທົດສອບໃນລະດັບນີ້ມີ 2 ວິທີຄື:
  - Black Box Testing
    - ເປັນວິທີການທົດສອບທີ່ມີການແບ່ງຂໍ້ມູນປ້ອນເຂົ້າອອກເປັນກຸ່ມດັ່ງນີ້
      1. ຄ່າຂໍ້ມູນຕ່ຳສຸດ
      2. ຄ່າຂໍ້ມູນສູງສຸດ
      3. ຄ່າຂໍ້ມູນທີ່ເປັນຕົວແທນກຸ່ມເປັນຄ່າຂໍ້ມູນທີ່ໃກ້ຄຽງກັບຄ່າກາງ
      4. ຄ່າຂໍ້ມູນເກີນຂອບເຂດຂອງຂໍ້ມູນຂອງເງື່ອນໄຂແຕ່ລະຊ່ວງ

ການທົດສອບການລວມຫົວໜ່ວຍຍ່ອຍ

- ເປັນການທົດສອບການເຮັດວຽກຂອງກຸ່ມໂປຣແກຣມ ຫຼື ທົດສອບການລວມໂປຣແກຣມຍ່ອຍເຂົ້າດ້ວຍກັນ ໂດຍເຮັດໜ້າທີ່ໃດໜຶ່ງຮ່ວມກັນ
- ເປັນການທົດສອບພາກສ່ວນສື່ສານການເຮັດວຽກຮ່ວມກັນລະຫວ່າງແຕ່ລະສ່ວນຍ່ອຍ
- ສ່ວນທີ່ຈະຖືກທົດສອບມີ 2 ຢ່າງຄື: ພາກສ່ວນສື່ສານ ແລະ ຜົນການເຮັດວຽກຂອງພາກສ່ວນລວມ
- ການທົດສອບແບບລວມສາມາດເຮັດໄດ້ 2 ລັກສະນະຄື: ແບບລວມໜ່ວຍທັງໝົດແລ້ວທົດສອບຄັ້ງດຽວ ແລະ ແບບເພີ່ມເທື່ອລະໂມດູນ

ການທົດສອບການລວມຫົວໜ່ວຍຍ່ອຍ

- ການທົດສອບແບບເພີ່ມເທື່ອລະໂມດູນມີ 2 ວິທີ
  - ການທົດສອບແບບເພີ່ມໂມດູນຈາກເທິງລົງລຸ່ມ (Top-down Appoach)
    - ເປັນການທົດສອບໂດຍເພີ່ມເທື່ອລະໂມດູນຈາກເທິງລົງລຸ່ມຕາມລຳດັບໂຄງສ້າງ ຄວບຄຸມ ໝາຍຄວາມວ່າ ໂມດູນທີ່ຢູ່ລະດັບເທິງຈະເອີ້ນໃຊ້ໂມດູນທີ່ຢູ່ລະດັບລຸ່ມ
    - ມີຫລັກການທົດສອບດັ່ງນີ້
      1. ຂະບວນການເຮັດວຽກໃດໜຶ່ງທີ່ຈະທົດສອບຈະຕ້ອງມີໂມດູນຫລັກເພື່ອຮັບຂໍ້ມູນທົດສອບແລ້ວສົ່ງຜ່ານໄປຫາໂມດູນທົດສອບ, ໂມດູນຫລັກເອີ້ນວ່າ Driver
      2. ຫາກໂມດູນທີ່ຈະຖືກທົດສອບຕ້ອງການໂມດູນຍ່ອຍຈຶ່ງຈະສົມບູນ ແຕ່ໃນຂະນະນັ້ນໂມດູນຍ່ອຍຍັງສ້າງບໍ່ແລ້ວ ທີມງານຈະຕ້ອງສ້າງ ໂມດູນແທນ (Stub Module) ຂຶ້ນມາແທນເພື່ອທົດສອບກ່ອນ
      3. ການທົດສອບຈະເຮັດທຸກຄັ້ງທີ່ມີການເພີ່ມໂມດູນ

ການທົດສອບການລວມຫົວໜ່ວຍຍ່ອຍ

- ການທົດສອບແບບເພີ່ມເທື່ອລະໂມດູນມີ 2 ວິທີ
  - ການທົດສອບແບບເພີ່ມໂມດູນຈາກລຸ່ມຂຶ້ນເທິງ (Bottom-up Approach)
    - ຈະທົດສອບໂດຍເລີ່ມຈາກໂມດູນລຸ່ມສຸດກ່ອນ ເປັນການລວມເອົາໂມດູນລຸ່ມສຸດເຂົ້າກັນເປັນກຸ່ມ (Cluster) ເພື່ອທົດສອບການເຮັດວຽກຮ່ວມກັນ
    - ຕ້ອງເຮັດ Driver ຂຶ້ນມາເພື່ອທົດສອບການເຮັດວຽກຂອງໂມດູນໃນລະດັບລຸ່ມ ເມື່ອທົດສອບຮຽບຮ້ອຍແລ້ວຈຶ່ງຖອດ Driver ອອກ ແລ້ວແທນທີ່ດ້ວຍ Cluster ໃໝ່ທີ່ເພີ່ມເຂົ້າມາ ເຮັດແນວນັ້ນໄປເລື້ອຍໆ
    - ເຄື່ອງມືທີ່ໃຊ້ແບ່ງ Cluster ໄດ້ດີທີ່ສຸດຄື Structure Chart
    - ການເລືອກວິທີການທົດສອບລະບົບລວມຂຶ້ນຢູ່ກັບໂຄງສ້າງຄວບຄຸມການເຮັດວຽກຂອງລະບົບ
      - ຖ້າເປັນລະບົບທີ່ຄວບຄຸມແລະຕັດສິນໃຈຢູ່ທາງເທິງໃຫ້ເລືອກ Top-down
      - ຖ້າເປັນລະບົບທີ່ການຄວບຄຸມຂຶ້ນຢູ່ກັບເງື່ອນໄຂທາງທຸລະກິດໃຫ້ເລືອກ Bottom-up

ການທົດສອບການລວມຫົວໜ່ວຍຍ່ອຍ

- ການທົດສອບແບບເພີ່ມເທື່ອລະໂມດູນມີ 2 ວິທີ
  - ບໍ່ວ່າຈະເລືອກວິທີໃດກໍ່ຕາມຫລັງຈາກການທົດສອບລະດັບລວມແລ້ວ ຈະຕ້ອງມີການເຮັດ Regression Testing ເພື່ອທົດສອບການເຮັດວຽກຂອງໂມດູນຊ້ຳອີກເທື່ອໜຶ່ງ ໂດຍສາມາດທົດສອບດ້ວຍກໍລະນີທົດສອບຊຸດເດີມອີກ
  - ແຕ່ເພື່ອເປັນການປະຢັດເວລາອາດຈະທົດສອບສະເພາະສ່ວນທີ່ເພີ່ມເຂົ້າມາ
  - ຖ້າຕ້ອງການທົດສອບຊ້ຳທັງໝົດອີກເທື່ອໜຶ່ງສາມາດໃຊ້ເຄື່ອງມື Capture/Playback Tools

ການທົດສອບລະບົບ

- ເປັນການທົດສອບການເຮັດວຽກຂອງລະບົບເມື່ອເອົາຊອບແວຣ໌ມາລວມເຂົ້າກັບອົງປະກອບອື່ນໆໄດ້ແກ່ ອຸປະກອນ, ບຸກຄະລາກອນ ແລະ ຂໍ້ມູນ
- ເພື່ອທົດສອບວ່າລະບົບເຮັດວຽກໄດ້ຖືກຕ້ອງຕາມຂໍ້ກຳໜົດ ແລະ ຄວາມຕ້ອງການຂອງຜູ້ໃຊ້
- ການທົດສອບລະບົບແບ່ງອອກເປັນ 2 ລັກສະນະ
  - Alpha and Beta Testing
  - Runtime Operation Testing

ການທົດສອບລະບົບ

- Alpha and Beta Testing
  - ບາງຄັ້ງເອີ້ນວ່າ ການທົດສອບການສົ່ງມອບ ຫຼື ການທົດສອບການຍອມຮັບ ຂອງລູກຄ້າ ເນື່ອງຈາກລູກຄ້າເປັນຜູ້ທົດສອບເອັງ
  - Alpha Testing ເປັນການທົດສອບລະບົບໂດຍຜູ້ໃຊ້ຢູ່ສະຖານທີ່ຜະລິດຊອບແວຣ໌ໂດຍຜູ້ໃຊ້ໃຊ້ງານພາຍໄຕ້ສະຖານະການຈຳລອງຂຶ້ນ
  - Beta Testing ເປັນການນຳເອົາຊອບແວຣ໌ໄປໃຫ້ຜູ້ໃຊ້ໄດ້ທົລອງໃຊ້ງານຊອບແວຣ໌ໃນສະຖານທີ່ຈິງດ້ວຍຕົນເອັງໂດຍບໍ່ມີທີມງານຄອຍສັງເກດ

ການທົດສອບລະບົບ

- Runtime Operation Testing
  - ເປັນການທົດສອບຂະນະທີ່ລະບົບກຳລັງເຮັດວຽກຢູ່
  - ສິ່ງທີ່ຕ້ອງທົດສອບມີດັ່ງນີ້
    - ທົສອບການກູ້ຄືນ (Recovery Testing)
    - ທົດສອບໃນກໍລະນີຂັບຂັນ (Stress Testing) ເປັນການທົດສອບໃນສະຖານະການບໍ່ປົກກະຕິ ໂດຍລະບົບຈະຕ້ອງເຮັດວຽກຕໍ່ໄປໄດ້
    - ທົດສອບສະມັດຕະພາບ (Performance Testing)
    - ທົດສອບການຮັກສາຄວາມປອດໄພ (Security Testing)
    - ການທົດສອບການເຮັດເອກະສານ (Document Testing)

ການທົດສອບຊອບແວຣ໌ແບບອັດຖຸ

- ການທົດສອບລະດັບຫົວໜ່ວຍຍ່ອຍ (Unit Testing)
  - ເອີ້ນວ່າ Class Testing ເນື່ອງຈາກວ່າ Class ຫຼື Object ໄດ້ລວມເອົາ ຂໍ້ມູນ ແລະ ພຶດຕິກຳໄວ້ນຳກັນ
  - ການທົດສອບ Class ຈະຕ້ອງພິຈະລະນາໃນສ່ວນດຳເນີນການແລະ ພຶດຕິກຳໃນແຕ່ລະສະຖານະການຂອງມັນ

ການທົດສອບຊອບແວຣ໌ແບບອັດຖຸ

- ການທົດສອບລະບົບລວມ (Integration Testing)
  - ວິທີທົດສອບແບບ Top-down ແລະ Bottom-up ແມ່ນບໍ່ສາມາດໃຊ້ໄດ້ກັບຊອບແວຣ໌ແບບ Object
  - ວິທີທົດສອບແບບລວມໜ່ວຍຂອງຊອບແວຣ໌ແບບ Object
    - Thread-based Testing ແມ່ນການລວມ Class ທີ່ຕອບສະໜອງຕໍ່ເຫດການດຽວກັນໄວ້ເປັນກຸ່ມດຽວກັນ ເອີ້ນວ່າ Thread ແລະ ແຕ່ລະ Thread ຈະຖືກທົດສອບເປັນອິດສະຫລະ
    - Use-based Testing ເປັນການທົດສອບທີ່ເລີ່ມຕົ້ນຈາກ Independent Class ຈາກນັ້ນຈຶ່ງເພີ່ມລະດັບຂຶ້ນໄປທີ່ Dependent Class ແຕ່ລະລະດັບຂຶ້ນໄປເລື້ອຍຈົນຄົບ

ກໍລະນີ ແລະ ການວາງແຜນການທົດສອບ

- ການອອກແບບກໍລະນີທົດສອບແມ່ນການກຳໜົດຊຸດຂໍ້ມູນເພື່ອປ້ອນເຂົ້າ (Input) ແລະ ຜົນໄດ້ຮັບທີ່ຄາດຫວັງ (Output) ໂດຍມີເປົ້າໝາຍເພື່ອຄົ້ນພົບຂໍ້ຜິດພາດ ແລະ ຂໍ້ບົກພ່ອງຂອງຊອບແວຣ໌ໃຫ້ຫລາຍທີ່ສຸດ
- ການວາງແຜນການທົດສອບເປັນການກຳໜົດລາຍລະອຽດການເຮັດວຽກໃນແຕ່ລະຂັ້ນຕອນຂອງຂະບວນການທົດສອບ ສິ່ງທີ່ໄດ້ຄື ເອກະສານແຜນການທົດສອບ (Test Plan)
- ແຜນການທົດສອບ ແມ່ນເອກະສານທີ່ປະກອບດ້ວຍຊຸດຂໍ້ມູນທີ່ປ້ອນເຂົ້າຂອງແຕ່ລະເສັ້ນທາງ ແລະ ຜົນຂອງການທົດສອບແຕ່ລະເສັ້ນທາງ

ກໍລະນີ ແລະ ການວາງແຜນການທົດສອບ

- ຕ້ອງກຳໜົດຮູບແບບເອກະສານສຳຫລັບຂຽນກໍລະນີທົດສອບແຕ່ລະກໍລະນີ ແລະ ຮູບແບບເອກະສານສຳຫລັບຂຽນແຜນທົດສອບ

ແບບຟອມ Test Case

Test Case Name:	Test Case ID:
Purpose of Test:	Testing Object: (Unit, Module, Application , Class)
Test Attribute:	
Test Focus: (Function, Feature, Interface, etc.)	
Test Type: (Alpha, Beta, Unit, Integration, System)	
Test Process:	ຄຳສັ່ງໃຊ້ທົດສອບໃນກໍ່ລະນີຕ່າງໆ ເລີ່ມຈາກສະຖານະການເລີ່ມຕົ້ນ, ຂໍ້ມູນນຳເຂົ້າ ແລະ ຜົນຄາດວ່າຈະໄດ້ຮັບ
Test Result:	ສະແດງຜົນຄາດວ່າຈະໄດ້ຮັບ, ຜົນຈາກການທົດສອບ ແລະ ປຽບທຽບຜົນທັງສອງ
Action:	ການແກ້ໄຂ ແລະ ຜົນຕອບຮັບຈາກການທົດສອບໃໝ່

ກໍລະນີ ແລະ ການວາງແຜນການທົດສອບ

- ຕ້ອງກຳໜົດຮູບແບບເອກະສານສຳຫລັບຂຽນກໍລະນີທົດສອບແຕ່ລະກໍລະນີ ແລະ ຮູບແບບເອກະສານສຳຫລັບຂຽນແຜນທົດສອບ

Project Name:_____Project ID:_____
Project Manager:_____QA Manager:_____

ແບບຟອມ Test Plan

Test									
Test ID		Test Name	1	2	3	…	n	Planned Date	
ID	Tester							Completed	Successful

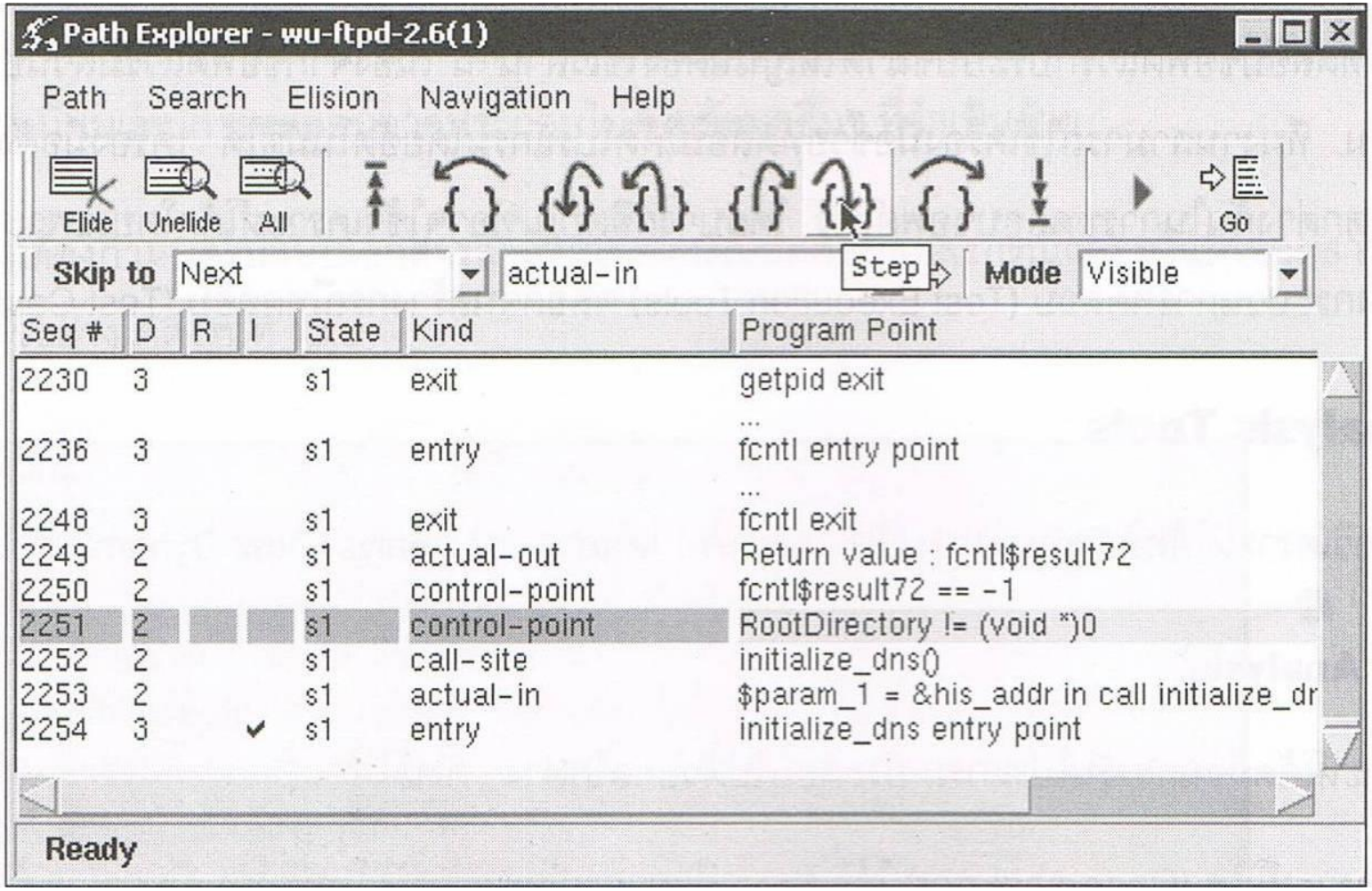
ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

- Code Analysis Tools
  - ເປັນເຄື່ອງມືທີ່ໃຊ້ວິເຄາະໂຄດໂປຣແກຣມ ແບ່ງອອກເປັນ 2 ປະເພດ
    - Static Analysis ໃຊ້ວິເຄາະໂຄດກ່ອນການ Run ໂປຣແກຣມ ປະກອບດ້ວຍກົນໄກຕ່າງໆດັ່ງນີ້
      - Code Analyzer ໃຊ້ວິເຄາະໄວຍະກອນ
      - Structure Checker ໃຊ້ສ້າງ Graph ສະແດງໂຄງສ້າງຄວບຄຸມ
      - Data Analyzer ໃຊ້ທົບທວນໂຄງສ້າງຂໍ້ມູນ, ການປະກາດໃຊ້ຂໍ້ມູນ ແລະ ການປະກາດ Interface
      - Sequence Checker ໃຊ້ກວດສອບລຳດັບຂອງເຫດການ

ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

- Code Analysis Tools
  - ເປັນເຄື່ອງມືທີ່ໃຊ້ວິເຄາະໂຄດໂປຣແກຣມ ແບ່ງອອກເປັນ 2 ປະເພດ
    - Dynamic Analysis ໃຊ້ວິເຄາະໂຄດໃນຂະນະ Run ໂປຣແກຣມ ຊ່ວຍໃຫ້ສາມາດເບິ່ງສະຖານະການຕ່າງໆໃນເວລາ Run ໂປຣແກຣມ
      - ບາງຄັ້ງເອີ້ນວ່າ Program Monitor
      - ສາມາດນັບຈຳນວນການເອີ້ນໃຊ້ຄອມໂພເນັ້ນຫຼືຈຳນວນແຖວໂຄດທີ່ກຳລັງ run ຢູ່ໄດ້ ຊ່ວຍໃຫ້ຮູ້ເຖິງເສັ້ນທາງການທົດສອບໂປຣແກຣມໄດ້ງ່າຍ

ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

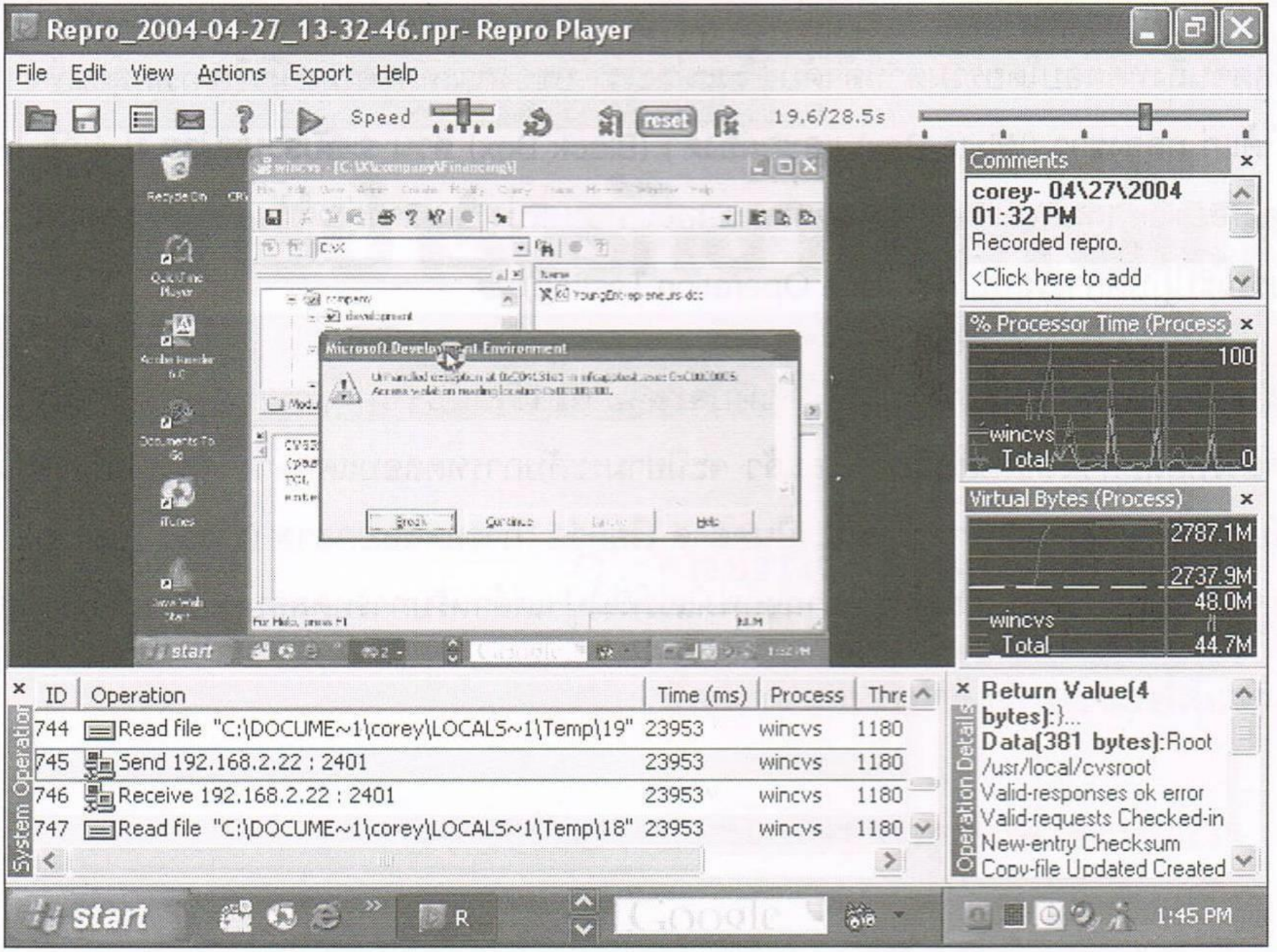


ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

- Test Execution Tools
  - ເປັນເຄື່ອງມືທີ່ໃຊ້ສ້າງຂະບວນການແບບອັດຕະໂນມັດ ຊ່ວຍໃນການວາງແຜນ ແລະ ດຳເນີນການທົດສອບໄດ້ງ່າຍຂຶ້ນ
  - ເຄື່ອງມືທີ່ເໝາະສົມກັບການທົດສອບໃນລັກສະນະນີ້ໄດ້ແກ່
    - Capture and playback Tool
    - Stub and Driver
    - Automated Testing Environment

ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

- Test Execution Tools
  - Capture and playback Tool
    - ໃຊ້ເບິ່ງ ຫຼື ບັນທຶກເຫດການການປ້ອນຂໍ້ມູນເຂົ້າ ແລະ ການຕອບສະໜອງຂອງໂປຣແກຣມ
  - Stub and Driver
    - ເປັນເຄື່ອງມືທີ່ຊ່ວຍໃຫ້ສາມາດປະສານການເຮັດວຽກຂອງ Stub ກັບ Driver ໄດ້ອັດຕະໂນມັດ
  - Automated Testing Environment
    - ສາມາດຮ່ວມກັບເຄື່ອງມືອື່ນໆເພື່ອໃຫ້ມີຄວາມສາມາດຫລາຍຂຶ້ນໄດ້
    - ສ່ວນຫລາຍຈະເຊື່ອມຕໍ່ກັບພາກສ່ວນທົດສອບຖານຂໍ້ມູນ, ເຄື່ອງມືວັດແທກຜົນໄດ້ຮັບ, ເຄື່ອງມືວິເຄາະໂຄດ, ໜ້າຕ່າງຂຽນໂຄດ, ເຄື່ອງມືສ້າງສະຖານະການແລະແບບຈຳລອງ ແລະ ອື່ນໆ


Repro_2004-04-27_13-32-46.rpr- Repro Player
File Edit View Actions Export Help
Speed
reset
19.6/28.5s
Comments
corey- 04\27\2004 01:32 PM
Recorded repro.
<Click here to add
% Processor Time (Process)
100
wincvs
Total
0
Virtual Bytes (Process)
2787.1M
2737.9M
48.0M
wincvs
Total
44.7M
Microsoft Development Environment
System Operation
ID
Operation
Time (ms)
Process
Thre
744 Read file "C:\DOCUME~1\corey\LOCALS~1\Temp\19" 23953 wincvs 1180
745 Send 192.168.2.22 : 2401 23953 wincvs 1180
746 Receive 192.168.2.22 : 2401 23953 wincvs 1180
747 Read file "C:\DOCUME~1\corey\LOCALS~1\Temp\18" 23953 wincvs 1180
Operation Details
Return Value(4 bytes):}...
Data(381 bytes):Root
/usr/local/cvsroot
Valid-responses ok error
Valid-requests Checked-in
New-entry Checksum
Copy-file Updated Created
start
R
1:45 PM

ເຄື່ອງມືການທົດສອບແບບອັດຕະໂນມັດ

- Test Case Generator
  - ເປັນເຄື່ອງມືສ້າງກໍລະນີທົດສອບແບບອັດຕະໂນມັດ ເພື່ອຊ່ວຍໃຫ້ສາມາດສ້າງກໍລະນີທົດສອບໄດ້ຄອບຄຸມທຸກສະຖານະການ ຫຼື ທຸກກໍລະນີ