

## ທ້າວ ນູຊິວ ເຮີ ຫ້ອງ 3CW1

### Homework5

### ສະຫຼຸດເນື້ອໃນບົດຮຽນ ບົດທີ5

### ການປະເມີນຕົ້ນທຶນຂອງຊອບແວ

1. ການປະເມີນຕົ້ນທຶນຂອງຊອບແວເປັນກິດຈະກຳທີ່ສຳຄັນທີ່ສຸດ ໃນການວາງແຜນໂຄງການ, ເປັນການປະມານຄ່າໃຊ້ຈ່າຍທີ່ເກີດຂຶ້ນທັງຫມົດໃນການຜະລິດ ຊອບແວເພື່ອເອົາມາເປັນຕົ້ນທຶນຂອງຊອບແວ ແລ້ວນຳໄປປະ ເມີນລາຄາຂອງຊອບແວ, ຄ່າໃຊ້ຈ່າຍທີ່ສຳຄັນທີ່ສຸດແມ່ນຄ່າແຮງງານ (Effort), ຄ່າໃຊ້ຈ່າຍ \_ ໃນການຊື້ວັດຖຸດິບຕ່າງໆ, ຕົ້ນທຶນຂອງໂຄງການແມ່ນຕົ້ນທຶນຂອງການຜະລິດຊອບ, ແວຣ໌ລວມກັບ ຕົ້ນທຶນອື່ນໆນຳ ຈ ບົວສິດ ໄຊຍະຈັກ, ຄວທ, ມຊ 2011-2012 ຕົ້ນທຶນຂອງໂຄງການ
2. ການປະມານຂະໜາດຂອງຊອບແວ ປະສິດທິຜົນຂອງການເຮັດວຽກສາມາດຄຳນວນໄດ້ຈາກຈຳນວນຂອງ ວຽກທີ່ເຮັດ (Size) ຫານດ້ວຍຈຳນວນເວລາທີ່ຕ້ອງການໃນການ ຜະລິດ (Effort) ຊຶ່ງອາດມີຫົວໜ່ວຍ ເປັນ Person-Hours, Man-Day, Man-Month

$$\text{Productivity} = \text{Size}/\text{Effort}$$

ການວັດແທກຂະໜາດຂອງຊອບແວນັ້ນມີ 2 ປະເພດຄື: 0 ນັບຈຳນວນແຖວຂອງໂຄດ ແລະ ນັບຈຳນວນ Function:

+ ການນັບຈຳນວນແຖວຂອງໂຄດ

- Simple Line Count ເປັນວິທີນັບໂຄດທຸກແຖວທີ່ມີຢູ່ໃນ Source File
- Physical Line (LINES) ບໍ່ນັບແຖວທີ່ເປັນນິຍາມຂອງຕົວປ່ຽນ
- Physical Line of Code : ບໍ່ນັບຈະນວນແຖວຫວ່າງແລະ comment
- Logical Lines of Code (LLOC) ຄ້າຍຄືກັນກັບ physical ແຕກຕ່າງຢູ່ບ່ອນວ່າ Logical ນັ້ນຈະ ນັບແຖວທີ່ມີການເຊື່ອມຕໍ່ກັນດ້ວຍເຄື່ອງໝາຍ ”\_” ເປັນແຖວດຽວກັນ
- Statements (STMT) ເປັນການນັບຈຳນວນປະໂຫຍກຄຳສັ່ງ

+ ການນັບຈຳນວນ Function (Function Point: FP)

- ເປັນການວັດແທກຂະໜາດຂອງຊອບແວຕາມຈຳນວນ function ຂອງໂປຣແກຣມຈາກຂໍ້ກຳໜົດ ຄວາມຕ້ອງການ
- ບໍ່ຂຶ້ນກັບພາສາຂຽນໂປຣແກຣມທີ່ເລືອກໃຊ້ ແລະ ການອອກແບບ
- ມີສູດດັ່ງນີ້:  $FP = UFP \times VAF$

-ຈາກສູດ, ຈຳນວນ Function ຄຳນວນໄດ້ຈາກຄ່າ FP ທີ່ບໍ່ທັນໄດ້ຖືກປັບແຕ່ງ (Unadjusted Function Point: UFP) ຄູນກັບຄ່າປັດໃຈ ຄູນລັກສະໂພງລະບົບ (Value Adjustment Factor: VAF)

3. ເທັກນິກການປະເມີນຕົ້ນທຶນ ແລະ ຄວາມພະຍາຍາມ ການປະເມີນຕົ້ນທຶນ ແລະ Effort ທີ່ດີຈະຕ້ອງໃຫ້ໄກ້ຄຽງກັບຄວາມເປັນຈິງຊຶ່ງເປັນການເຮັດໄດ້ຍາກ ສະນັ້ນ ຈຶ່ງໄດ້ມີການຄິດຄົ້ນເທັກນິກການປະເມີນຂຶ້ນມາຫຼາຍແບບດັ່ງນີ້

+ Algorithmic Cost: ການໃຊ້ແບບຈຳລອງທາງຄະນິດສາດເພື່ອປະເມີນໂດຍແບບຈຳລອງນັ້ນ ຖືກພັດທະນາ Modeling ມາຈາກການລວບລວມຂໍ້ມູນຕົ້ນທຶນຈິງໃນອະດີດທີ່ມີຄວາມສຳພັນກັບການວັດແທກ ບາງຢ່າງຂອງຊອບແວ ເຊັ່ນ: ຂະໜາດຂອງມັນ

+Expert Judgement ການໃຊ້ຄວາມເຫັນຜູ້ຊ່ຽວຊານໃນການປະເມີນ ປຽບທຽບກັບຂໍ້ມູນໃນອາດີດ ເພື່ອປຶກສາ ແລະ ຕົກລົງກຳໜົດຕົ້ນທຶນຮ່ວມກັນ

+Estimation by Analogy ການປະເມີນດ້ວຍການວິເຄາະ ໂດຍອາໄສຂໍ້ມູນຈາກໂຄງການໃນທຸລະກິດດຽວກັນທີ່ | ເຮັດປະສົບຜົນສຳເລັດມາແລ້ວເປັນຂໍ້ມູນຫຼັກໃນການວິເຄາະ

+Parkinson's Law | ເປັນການແຈກຢາຍວຽກໃຫ້ກັບບຸກຄະລາກອນຕາມໄລຍະເວລາທີ່ມີຢູ່

+Pricing to Win | ການປະເມີນເພື່ອໃຫ້ຊະນະການປະມຸນ

+ເທັກນິກການປະເມີນແບບ COCOMO ໄດ້ຖືກພັດທະນາເປັນລຸ້ນທີ 2 ໃນປີ 1997 ໂດຍລວບລວມຂໍ້ມູນຈາກໂຄງການທັງໝົດ 161 ໂຄງການ ໆ COCOMO II ໄດ້ແບ່ງແບບຈຳລອງອອກເປັນ 3 ຊະນິດເພື່ອປະເມີນຕາມໄລຍະຕ່າງໆຂອງພັດທະນາຊອບແວ:

- Application-Composition Model ເໝາະສົມກັບການຜະລິດຊອບແວແບບ component ແລະໃຊ້ຢູ່ໃນໄລຍະສະຫຼຸບ concept ໃນການດຳເນີນງານ, ໃຊ້ Object Point ແທນຂະໜາດຂອງຊອບແວ

- Early Design Model ໃຊ້ປະເມີນຢູ່ໃນໄລຍະກ່ອນອອກແບບຊອບແວ ຫຼັງຈາກການກຳໜົດຄວາມຕ້ອງການ, ໃຊ້ FP

- Post-architecture Model ໃຊ້ຫຼັງການອອກແບບ