ທ້າວ ນູຊົ່ວ ເຮີ ຫ້ອງ 3CW1

Homework5

ສະຫຼຸດເນື້ອໃນບົດຮຽນ ບົດທີ5

ການປະເມີນຕົ້ນທືນຂອງຊອບເເວຣ

1. ການປະເມີນຕົ້ນທືນຂອງຊອບແວຣ໌ເປັນກິດຈະກໍາທີ່ສໍາຄັນທີ່ສຸດ ໃນການວາງແຜນໂຄງການ,ເປັນການປະມານຄ່າໃຊ້ຈ່າຍທີ່ເກີດຂື້ນທັງຫມົດໃນການຜະລິດ ຊອບແວຣ໌ເພື່ອເອົາມາເປັນຕົ້ນທືນຂອງຊອບແວຣ໌ ແລ້ວນໍາໄປປະ ເມີນລາຄາຂອງຊອບແວຣ໌, ຄ່າໃຊ້ຈ່າຍທີ່ສໍາຄັນທີ່ສຸດແມ່ນຄ່າແຮງງານ (Effort), ຄ່າໃຊ້ຈ່າຍ \_ ໃນການຊື້ວັດຖຸດິບຕ່າງໆ, ຕົ້ນທືນຂອງໂຄງການແມ່ນຕົ້ນທືນຂອງການຜະລິດຊອບ, ແວຣ໌ລວມກັບຕົ້ນທືນອື່ນໆນໍາ ຈ ບົວສົດ ໄຊຍະຈັກ, ຄວທ, ມຊ 2011-2012ຕົ້ນທືນຂອງໂຄງການ
2. ການປະມານຂະໜາດຂອງຊອບແວຣ ປະສິດທິຜົນຂອງການເຮັດວຽກສາມາຄໍານວນໄດ້ຈາກຈໍານວນຂອງວຽກທີ່ເຮັດ (Size) ຫານດ້ວຍຈໍານວນເວລາທີ່ຕ້ອງການໃນການ ຜະລິດ (Effort) ຊຶ່ງອາດມີຫົວຫນ່ວຍເປັນ Person-Hours, Man-Day, Man-Month

Productivity = Size/Effort

ການວັດແທກຂະຫນາດຂອງຊອບແວຣ໌ນັ້ນມີ 2 ປະເພດຄື: o ນັບຈໍານວນແຖວຂອງໂຄດ ແລະ ນັບຈໍານວນ Function:

+ ການນັບຈໍານວນແຖວຂອງໂຄດ

-Simple Line Count ເປັນວິທີນັບໂຄດທຸກແຖວທີ່ມີຢູ່ໃນ Source File

-Physical Line (LINES) ບໍ່ນັບແຖວທີ່ເປັນນິຍາມຂອງຕົວປ່ຽນ

-Physical Line of Code : ບໍ່ນັບຈະນວນເເຖວຫວ່າງແລະ comment

-Logical Lines of Code (LLOC) ຄ້າຍຄືກັນກັບ physical ແຕກຕ່າງຢູ່ບ່ອນວ່າ Logical ນັ້ນຈະນັບແຖວທີ່ມີການເຊື່ອມຕໍ່ກັນດ້ວຍເຄື່ອຫມາຍ ”\_” ເປັນແຖວດຽວກັນ

-Statements (STMT) ເປັນການນັບຈໍານວນປະໂຫຍກຄໍາສັ່ງ

+ ການນັບຈໍານວນ Function (Function Point: FP)

-ເປັນການວັດແທກຂະຫນາດຂອງຊອບແວຣ໌ຕາມຈໍານວນ function ຂອງໂປຣແກຣມຈາກຂໍ້ກໍາຫນົດຄວາມຕ້ອງການ

- ບໍ່ຂື້ນກັບພາສາຂຽນໂປຣແກຣມທີ່ເລືອກໃຊ້ ແລະ ການອອກແບບ

- ມີສູດດັ່ງນີ້: FP = UFP x VAF

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

1. ເທັກນິກການປະເມີນຕົ້ນທືນ ແລະ ຄວາມພະຍາຍາມ ການປະເມີນຕົ້ນທືນ ແລະ Effort ທີ່ດີຈະຕອ້ງໃຫ້ໄກ້ຄຽງກັບຄວາມເປັນຈິງຊຶ່ງເປັນການເຮັດໄດ້ຍາກ ສະນັ້ນ ຈຶ່ງໄດ້ມີການຄິດຄົ້ນເທັກນິກການປະເມີນຂື້ນມາຫຼາຍແບບດັ່ງນີ້

+ Algorithmic Cost: ການໃຊ້ແບບຈໍາລອງທາງຄະນິດສາດເພື່ອປະເມີນໂດຍແບບຈໍາລອງນັ້ນຖືກພັດທະນາ Modelingມາຈາກການລວບລວມຂໍ້ມູນຕົ້ນທືນຈິງໃນອະດີດທີ່ມີຄວາມສໍາພັນກັບການວັດແທກ ບາງຢ່າງຂອງຊອບແວຣ໌ ເຊັ່ນ: ຂະຫນາດຂອງມັນ

+Expert Judgement ການໃຊ້ຄວາມເຫັນຜູ້ຊ່ຽວຊານໃນການປະເມີນ ປຽບທຽບກັບຂໍ້ມູນໃນອາດີດ ເພື່ອປຶກສາ ແລະ ຕົກລົງກໍາຫນົດຕົ້ນທືນຮ່ວມກັນ

+Estimation by Analogy ການປະເມີນດ້ວຍການວິເຄາະ ໂດຍອາໃສຂໍ້ມູນຈາກໂຄງການໃນທຸລະກິດດຽວກັນທີ່ | ເຮັດປະສົບຜົນສໍາເລັດມາແລ້ວເປັນຂໍ້ມູນຫລັກໃນການວິເຄາະ

+Parkinson's Law | ເປັນການແຈກຢາຍວຽກໃຫ້ກັບບຸກຄະລາກອນຕາມໄລຍະເວລາທີ່ມີຢູ່

+Pricing to Win | ການປະເມີນເພື່ອໃຫ້ຊະນະການປະມູນ

+ເທັກນິກການປະເມີນແບບ COCOMO ໄດ້ຖືກພັດທະນາເປັນລຸ້ນທີ່ 2 ໃນປີ 1997 ໂດຍລວບລວມຂໍ້ມູນຈາກໂຄງການທັງຫມົດ 161ໂຄງການ ຯ COCOMO II ໄດ້ແບ່ງແບບຈໍາລອງອອກເປັນ 3 ຊະນິດເພື່ອ ປະເມີນຕາມໄລຍະຕ່າງໆຂອງພັດທະນາຊອບແວຣ໌:

- Application-Composition Model ເຫມາະສົມກັບການຜະລິດຊອບແວຣ໌ແບບcomponent ແລະໃຊ້ຢູ່ໃນໄລຍະສະຫລຸບ concept ໃນການດໍາເນີນງານ, ໃຊ້ Object Point ແທນຂະຫນາດຂອງຊອບແວຣ໌

- Early Design Model ໃຊ້ປະເມີນຢູ່ໃນໄລຍະກ່ອນອອກແບບຊອບແວຣ໌ ຫລັງຈາກການກໍາຫນົດຄວາມຕ້ອງການ, ໃຊ້ FP

- Post-architecture Model ໃຊ້ຫລັງການອອກແບບ