

# Review SE

## 1. ដូចម្តេចដែលហៅថា Software Engineering ?តើSoftware Engineering មានផលប្រយោជន៍អ្វីខ្លះដល់អ្នកសិក្សា?

- Software Engineeringគឺជាបច្ចេកវិទ្យាមួយដែលភ្ជាប់មកជាមួយនូវក្បួនប្រព័ន្ធសាស្ត្រនិងTool ដែលយកមកDevelop Software ក្នុងគោលបំណងដើម្បីDevelop និងMaintenance Software អោយមានប្រសិទ្ធភាពខ្ពស់។

### ❖ ផលប្រយោជន៍នៃSoftware Engineeringសម្រាប់អ្នកសិក្សាគឺធ្វើអោយយើង៖

- សំរេចនូវការបង្កើតSoftwareមួយអោយយល់បានSuccessនិងទទួលបានResult ល្អ
- អាចសិក្សាកំហុសឆ្គងកន្លងមកដែលធ្លាប់កើតមាន
- ទទួលបានជោគជ័យអំពីការងារដែលធ្លាប់កើតបានធ្វើពីមុនមក
- មានDocument ច្បាស់លាស់ ការពារការមានបញ្ហាផ្សេងៗ
- ស្គាល់Model ច្រើនក្នុងការយកទៅប្រើប្រាស់តាមប្រភេទនៃSoftware
- ដឹងអំពីFlow ក្នុងការDevelop Software Productមួយ
- អាចប៉ាន់តម្លៃ រយៈពេល និងធនធានផ្សេងៗរបស់។

## 2. ចូរនិយាយពីគុណសម្បត្តិនៃការប្រើប្រាស់Software Engineering ក្នុងការDevelop Software?

### ❖ គុណសម្បត្តិនៃការប្រើSoftware Engineering ក្នុងការDevelop Softwareមាន៖

- Improved quality
- Improved productivity
- Improved reliability
- Improved requirement specification
- Improved cost and schedule estimation
- Better use of automated tool and techniques
- Less defect in final product
- Better maintenance of delivered software
- Well defined process

## 3. តើអ្វីទៅជាRequirement Elicitation? តើគេប្រើវិធីសាស្ត្រអ្វីខ្លះក្នុងការធ្វើ Requirement Elicitation?ចូរលើកឧទាហរណ៍មកបញ្ជាក់។

- Requirement Elicitation (gathering) គឺជាការសិក្សាទៅលើSystem ដើម្បីCollect Data អោយបានប្រាកដប្រជាពីUser, Customer ឬក៏ អ្នកដែលពាក់ព័ន្ធផ្សេងៗទៀត (other stakeholders) ហើយយកData ទាំងអស់នោះទៅAnalyze សម្រាប់បង្កើតSystem ឬProductមួយ។

### ❖ វិធីសាស្ត្រ ក្នុងការធ្វើ Requirement Elicitationរួមមាន៖

- Interviews ធ្វើការសម្ភាសអ្នកដែលពាក់ព័ន្ធ
- Questionnaires កម្រងសំណួរទាក់ទងSurvey
- User Observation ធ្វើការសង្កេតទៅលើUser

- Workshops បង្កើតការប្រជុំ
- Sample ឯកសារគំរូ
- Prototyping ប្រើប្រាស់Prototyping ដែលមានស្រាប់ ដើម្បីGathering data។

Example:

#### 4. ដើម្បីបង្កើតSoftwareដែលមានគុណភាព(Quality) តើត្រូវពឹងផ្អែកលើកត្តាអ្វីខ្លះ?

❖ ដើម្បីបង្កើតSoftwareដែលមានគុណភាព(Quality) យើងត្រូវពឹងផ្អែកលើកត្តាដូចខាងក្រោម៖

- Correctness លក្ខណៈត្រឹមត្រូវនិងសុក្រិតនៃSoftware
- Usability ងាយស្រួលរៀន ធ្វើការបានល្អ
- Integrity ការការពាររាយការណ៍ពីខាងក្រៅ
- Efficiency ធ្វើការលឿនមិនចំណាយCPU និងMemoryច្រើន
- Reliability ធ្វើបានត្រឹមត្រូវទៅតាមរូបភាពដោយមិនមានភាពError កើតឡើង
- Maintainability ងាយស្រួលក្នុងការកែប្រែនិងថែទាំ
- Reusability អាចយកផ្នែកណាមួយទៅប្រើប្រាស់លើProjectផ្សេងៗ។

#### 5. តើEvolutionary Software process model គឺជាអ្វី?ហើយមានលក្ខណៈយ៉ាងដូចម្តេច?

- Evolutionary Software process model គឺជាModel មួយដែលប្រើប្រាស់តែរបស់មានស្រាប់ វាច្រើនប្រើសម្រាប់លក្ខណៈSystem Maintenance។

❖ Evolutionary Software process model មានលក្ខណៈដូចជា៖

- Software ពីមួយជំនាន់ទៅមួយជំនាន់កាន់តែធំទៅៗ
- គុណភាពកាន់តែល្អជាងមុន
- កាន់តែស្មុគស្មាញជាងមុន
- រក្សាអ្វីដែលមានស្រាប់
- អាចលុបModuleណាមួយដែលលែងប្រើ
- បន្ថែមនូវអ្វីដែលថ្មី

#### 6. ចូរនិយាយពីផលប្រយោជន៍នៃការវាស់SoftwareតាមបែបMatric។ តើអ្វីទៅជា Software project matric? ចូរប្រៀបធៀប plan VS actual cost។

❖ ផលប្រយោជន៍នៃការវាស់SoftwareតាមបែបMatricរួមមាន៖

- អាចដឹងថាSoftware នោះមានគុណភាពល្អឬអត់
- អាចដឹងថាsoftware មានទំហំប៉ុណ្ណា
- អាចដឹងថាsoftwareប្រើប្រាស់លុយអស់ប៉ុណ្ណា
- អាចដឹងថាsoftwareនោះត្រូវការរយៈពេលប៉ុណ្ណា
- Software Project matric គឺជា Matric មួយដែលគេប្រើសម្រាប់វាស់ទៅលើProjectជាក់ស្តែងដែលវាវាស់ទៅលើ Effort, Productivity in FP, Cost, Size, Defects, និង Testing ។

❖ ប្រៀបធៀប Plan VS actual cost

- Plan Cost គឺជាតម្លៃដែលគេ សិក្សាឬប៉ាន់ក្នុងគំរោង។
- Actual cost គឺជាតម្លៃដែលគេចំណាយអស់ពិតពេលធ្វើSystemឡើង។

➢ លក្ខណៈដូចគ្នា៖

- ✓ ប៉ាន់តម្លៃរបស់Projectដូចគ្នា

➢ លក្ខណៈខុសគ្នា៖

Plan Cost	Actual Cost
តម្លៃអាចច្រើនឬតិចជាងActual Cost ព្រោះអាចមាន Problem កើតជាយថាហេតុ	តម្លៃអាចច្រើនឬតិចជាងPlan Cost ព្រោះអាចមាន Problem កើតជាយថាហេតុ

✚ ជាទូទៅ៖

- បើ Actual Value(AV) > Planned Value (PV) => អត់ល្អ=> Controlឡើងវិញទៅលើការងារគ្រប់គ្រង=> កែលំអ (Project ក្រោយៗ)
- បើ Actual Value(AV) < Planned Value (PV)=> ល្អ

✚ ដើម្បីរក%នៃការកើនឡើងរវាងAV និងPV គឺ  $((AV-PV)/PV)*100$

7. ដូចម្តេចដែលហៅថាDefect? មានប៉ុន្មានប្រភេទ? លើកឧទាហរណ៍បញ្ជាក់។

Defect គឺជាError ដែលកើតឡើងនៅក្នុងSystem របស់យើង។ Defectមាន3ប្រភេទគឺ៖

- Server គឺជាError មួយដែលប៉ះពាល់ធ្ងន់ធ្ងរអាចធ្វើអោយProductយើងបណ្តាលមកពីError Function ឬ Module ណាមួយ។

Example: ធ្វើអោយSystemទាំងមូលHang មិនអាចMoveទៅមុខរួច។

- Major គឺជាError មួយដែលប៉ះពាល់មធ្យមដល់ Productបណ្តាលមកពីLogicity ។

Example: Result ដែលទទួលបានពីSystemពុំត្រឹមត្រូវ100%។

- Minor គឺជាError មួយដែលប៉ះពាល់ស្រាលដល់Productមិនមានបញ្ហាក្នុងការដំណើរការរបស់Productទេ។

Example: Alert Message ថាSystem របស់យើងOut of date។

8. សម្មតិកម្មគេអោយ៖

- ចំនួនFunction Point សរុបនៃSystem
- Productivity ជាមធ្យម
- អត្រាថ្លៃឈ្នួលពលកម្ម(Labor Rate)ក្នុងមួយខែ
- ចំនួនមនុស្សក្នុងការបង្កើតSystem
- Average: ចំនួនLOC ជាមធ្យមក្នុងមួយFunction Point ទៅតាមProgramming language ដែលគេអោយ

### ក្នុងនាមយើងជាអ្នកគ្រប់គ្រងតើយើងអាចគណនាកម្ចីបានខ្លះ?

❖ សម្រាប់អ្នកគ្រប់គ្រងអាចគណនាកម្ចីបានដូចជា៖

➢ តម្លៃក្នុងមួយLOC៖

$$\text{Cost per LOC} = (\text{Labor rate/productivity})$$

➢ ចំនួនLOC សរុប៖

$$\text{Total number of LOC} = (\text{Function Point} / \text{ចំនួនLOC ជាមធ្យមក្នុង1Function})$$

➢ តម្លៃនៃProject៖

$$\text{Total cost of Project} = \text{Total number of LOC} * \text{Cost per LOC}$$

$$\text{Or} = \text{Effort} * \text{Labor rate}$$

➢ Effort តម្លៃពលកម្ម៖

$$\text{Effort} = (\text{Total cost of Project} / \text{Labor rate})$$

$$\text{Or} = (\text{Total cost of LOC} / \text{Productivity})$$

➢ រយៈពេលនៃការធ្វើProject៖

$$\text{Duration} = (\text{Effort} / \text{People})$$

### 9. ចូរនិយាយពីការវិភាគតម្រូវការនៃSoftwareតាមបែបFAST(Facilitated Application Specification Technique)? ហើយហេតុអ្វីបានជាFAST៖

- គេត្រូវជ្រើសរើសកន្លែងប្រជុំដែលមានអព្យាក្រឹតមួយ

- អ្នកសំរេចសំរួលនៃកិច្ចប្រជុំអាចជ្រើសរើសមកពីភាគីខាងក្រៅដែលមិនមែនជាភាគីដែលពាក់ព័ន្ធនៃកិច្ចប្រជុំ?

❖ ការវិភាគតម្រូវការនៃSoftwareតាមបែបFAST៖

➢ ការប្រជុំធ្វើឡើងនៅទីកន្លែងអព្យាក្រឹត

➢ ត្រូវបង្កើតច្បាប់នៃការរៀបចំនិងការចូលរួម

➢ ត្រូវមានAgenda(របៀបរឺ៖)ដែលមានគ្រប់ចំណុចសំខាន់ទាំងអស់

➢ មានFacilitator ដែលអាចជាCustomer Developer Outsider ដើម្បីមកគ្រប់គ្រងMeeting

➢ គោលបំណងគឺដើម្បីកំណត់ចេញនូវបញ្ហា ធាតុនៃដំណោះស្រាយ ពិភាក្សាអំពីចំណុចខុសគ្នា

❖ ហើយបានជាFAST៖

➢ គេត្រូវជ្រើសរើសកន្លែងប្រជុំដែលមានអព្យាក្រឹតមួយ ព្រោះថាដើម្បីកុំអោយមានភាគីខាងណាមានសម្ពាធរឺមានប្រៀបជាង។

➢ អ្នកសំរេចសំរួលនៃកិច្ចប្រជុំអាចជ្រើសរើសមកពីភាគីខាងក្រៅដែលមិនមែនជាភាគីដែលពាក់ព័ន្ធនៃកិច្ចប្រជុំ ព្រោះដើម្បីកុំអោយអ្នកសំរេចសំរួលមានភាពលំអៀងទៅខាងណា។

### 10. ចូរពន្យល់ពីគុណសម្បត្តិនិងគុណវិបត្តិនៃPrototyping Model។

➢ Prototyping ModelគឺជាModelមួយដែលប្រើប្រាស់នូវគំរូក្នុងផ្នែកខ្លះរឺផ្នែកថ្មីដែលយើងជាDeveloperមិនយល់អំពីបំណងរបស់Customer ត្រូវការ។

➤ គុណសម្បត្តិនៃPrototyping Model៖

- ងាយស្រួលនិងលឿនក្នុងការCollect Client Requirements
- Clients អាចចូលរួមផ្តល់Requirement បន្ថែមនិងបំពេញFeedbackផ្សេងៗ
- Relationship រវាងUser និងSystem មានលក្ខណៈងាយស្រួល
- Developer ងាយស្រួលកំណត់Domain ទៅលើApplication System។

➤ គុណវិប្បត្តិនៃPrototyping Model៖

- ជាModelមួយដែលប្រើប្រាស់បានតែSystemដំបូងប៉ុណ្ណោះ
- ត្រូវការCustomer និងDeveloper ធ្វើការជំនុំគ្នា
- Developer ប្រើប្រាស់Prototype សុទ្ធសាធដើម្បីផលិតSoftware Product

11. តើលោក Devis បានណែនាំដូចម្តេចខ្លះដើម្បីធ្វើTesting?

12. គេមានគម្រោង3ដូចខាងក្រោម៖

	Number Defect	Size in Function Point
Project1	20	200
Project2	30	100
Project3	40	800

a. គណនាដង់ស៊ីតេProject នីមួយៗ

b. តាមរយៈដង់ស៊ីតេក៏Project មួយណាមានគុណភាពល្អជាងគេ?

Solution:

5.1 គណនាដង់ស៊ីតេProject នីមួយៗ

តាមរូបមន្ត៖  $\text{Defect Density} = \frac{\text{Total Defect}}{\text{Total Size in Function Point}}$

⇒ Defect Density of Project1 =  $20/200=0.1$

⇒ Defect Density of Project1 =  $30/100=0.3$

⇒ Defect Density of Project1 =  $40/800=0.05$

5.2 រកProject មួយណាមានគុណភាពល្អជាងគេ

ដោយ  $\text{DensityP2} > \text{DensityP1} > \text{DensityP3}$

⇒ Project ទី3មានគុណភាពល្អជាងគេព្រោះថា Defect Densityមានតូចជាងគេ។

1. ឧបមារថាគេមានគម្រោងដូចខាងក្រោម៖

- ចំនួនLine of Codeប្រហែល240000LOC
- ចំនួនProductivity មធ្យម800LOC/Person
- តម្លៃពលកម្ម(ប្រាក់ខែ)500\$/Per-month

### 1.1 គណនាEffort()

### 1.2 គណនាតម្លៃសម្រាប់Project (Cost)។

Solution:

#### 6.1 គណនាEffort

❖ បម្រាប់ គេមាន៖

- ចំនួនLOC សរុប=240000LOC
- ផលិតភាពជាមធ្យម= 800LOC/Person-month

⇒  $\text{Effort} = (240000/800) = 300$ នាក់

#### 1.3 គណនាតម្លៃសម្រាប់Project (Cost)

❖ បម្រាប់ គេមាន៖

- ដោយក្នុងមួយខែម្នាក់ៗត្រូវចំណាយអស់USD 500\$/Per-month
- ហើយDeveloper ទាំងអស់មានចំនួន300នាក់

⇒ តម្លៃសរុបដើម្បីផលិតSoftware Product គឺ=  $500 \times 300 = 150,000$ USD។

## 13. លំហាត់

ដើម្បីបង្កើតSoftware Product មួយគេបានប៉ាន់ត្រូវការ line of Code (LOC) ប្រមាណ250000។ គេដឹងថា៖

ផលិតភាពជាមធ្យម(Average of Productivity) ប្រមាណ1250LOC/Person-Monthកម្លាំងពលកម្មសម្រាប់Developer ជាមធ្យមគឺUSD300/month

a) គណនាEffort (ចំនួនDeveloper ក្នុងមួយខែ) ផលិតអោយចប់Software Product នេះ។

b) តើគេត្រូវការចំណាយអស់ប៉ុន្មានដើម្បីបង្កើតSoftware Product នេះ?

c) បើគេត្រូវការប្រើប្រាស់Software Product ក្នុងរយៈពេល១០ថ្ងៃតើគេត្រូវចំណាយDeveloper ប៉ុន្មានអ្នក?

❖ ចម្លើយ

a) គណនាEffort (ចំនួនDeveloper ក្នុងមួយខែ) ផលិតអោយចប់Software Product នេះ

បម្រាប់ គេមាន៖

- ចំនួនLOC សរុប=250000LOC
- ផលិតភាពជាមធ្យម= 1250LOC/Person-month

⇒  $\text{Effort} = (250000/1250) = 200$ នាក់

b) ការចំណាយសរុប ដើម្បីបង្កើតSoftware Product នេះ

បម្រាប់ គេមាន៖

- ដោយក្នុងមួយខែម្នាក់ៗត្រូវចំណាយអស់USD 300.00

⇒ កុំខ្លាចការប្រឡងសង ព្រោះវាជារឿងដ៏សប្បាយរីករាយនៅក្នុងឆាកជីវិតរបស់មនុស្ស។

Latest Version

- ហើយDeveloper ទាំងអស់មានចំនួន200នាក់

⇒ តម្លៃសរុបដើម្បីផលិតSoftware Product គឺ=  $200 \times 300 = 60,000.00 \text{ USD}$ ។

c) ចំនួនDeveloper សរុបដើម្បីបង្កើតSoftware Product ក្នុងរយៈពេល១០ថ្ងៃ

បម្រាប់ គេមាន៖

- បើ3០ថ្ងៃត្រូវការDeveloper 200នាក់

- ហេតុនេះ1ថ្ងៃគេត្រូវការDeveloper  $200 \times 30 = 6000$ នាក់

⇒ ដូច្នេះបើ1០ថ្ងៃនោះគេត្រូវការDeveloper ចំនួន=  $6000 / 10 = 600$ នាក់។

**មើមើលហើយអត់ត្រូវ ឬប្រឡងធ្លាក់ មិនបាច់ជួលឃាតករធ្វើអីទេណាអូនខ្លាច  
ហើយមើត្រូវឬប្រឡងជាប់ ហៅថ្វីកស៊ីផងណាអូនឃ្លាន**

OK?