

សំណួរត្រៀមប្រឡង S1

I. SE Introduction :

១. SW (Software) : ជា Computer programs និង documentation ដែលពាក់ព័ន្ធ ។ Documentations ទាំងនោះមានដូចជា : Requirements, Design models និង User manuals ។

២. Software products ចែកចេញជា ២ សំខាន់ៗគឺ :

- Generic Software : បង្កើតឡើងដើម្បីលក់អោយអតិថិជនទូទៅ អាចយកទៅប្រើប្រាស់បាន ។
- Custom Software (Bespoke) : បង្កើតឡើងដើម្បីលក់អោយអតិថិជនពិសេសណាមួយទៅតាម Specification ជាក់លាក់ដែលគេកំណត់ ។
- Software Product អាចបង្កើតឡើងដោយ : Developing new programs, Configuring generic software systems, និង Reusing existing software ។

៣. SE (Software Engineering) : គឺជាការបង្កើត Software ដោយប្រើប្រាស់វិធីសាស្ត្រ និង ព្រំដែនបានត្រឹមត្រូវ ។ ដើម្បីបង្កើត Software បានល្អ Software Engineers ត្រូវប្រើប្រាស់យន្តការ (នីតិវិធី ដើម្បីសំរេចដល់គោលដៅ) ដោយប្រើប្រាស់ Tool (Programming, Technique, Constraint(Scope)) និង Resource (Budget, Time, People (Material people, Expert people)) អោយបានត្រឹមត្រូវ ។

៤. ដើម្បីបង្កើត Software ដែលមាន Quality ត្រូវពឹងផ្អែកលើកត្តាដូចខាងក្រោម :

- Usability : ងាយស្រួលរៀន, ធ្វើការបានល្អ ។
- Efficiency : ធ្វើការលឿន មិនចំនាយ CPU និង Memory ច្រើន ។
- Reliability : ធ្វើការបានត្រឹមត្រូវទៅតំរូវការ ដោយមិនមានភាព Error កើតឡើង ។
- Maintainability : ងាយស្រួលនៅក្នុងការកែប្រែ ។
- Reusability : អាចយកផ្នែកណាមួយទៅប្រើប្រាស់លើ Project ផ្សេងៗ ។

៥. Good Software Quality មានការពេញចិត្តដោយ :

- Customer : ដោះស្រាយបញ្ហាអោយគាត់ , មានតម្លៃសមរម្យ , និង ប្រើប្រាស់ Resource តិចនៅពេល Operation ។
- User : ងាយស្រួលរៀន, ងាយប្រើ, ធ្វើការលឿន ។
- Developer : ងាយ Design, ងាយ Maintenance, ងាយប្រើប្រាស់ Code ឡើងវិញ ។
- Manager : លក់បានច្រើន, មានការពេញចិត្តពីអតិថិជនទៅលើតម្លៃ Develop និង Maintenance ។

៦. Comparision :

- Computer Science: ជាការសិក្សា Theory និង Fundamental ដែលទាក់ទងនឹង Computer Hardware និង Software ។
- Software engineering : ជាផ្នែកមួយរបស់ Computer Science ដែលទាក់ទងនឹងការអភិវឌ្ឍន៍ និង ស្វែងរក Software ដែលមានប្រយោជន៍ ។
- System engineering : ជាការសិក្សារបស់បណ្តាញរបស់ការអភិវឌ្ឍន៍ Computer based System ដែលមានដូចជា : Hardware, Software, and Process engineering ។ System engineers ទាក់ទងទៅនឹង System Specifition, Architectural design, Integration, និង Development ។
- Software engineering : ជាផ្នែកមួយនៃ Process ដែលទាក់ទងទៅនឹងការអភិវឌ្ឍន៍នៃ Software infrastructure, Control, Application និង Database ដែលមាននៅក្នុង System ។

៧. Software Characteristic មានដូចជា :

- Software មិនមានរោងចក្រផលិតទេ ។ វាត្រូវតែបង្កើតដោយ Software engineers ។
- Software មិនខូចខាត រឺ ស៊ីករេចរិលដោយការលើកដាក់ រឺ ជូតឡើយ ។
- Software មិនមានគ្រឿងបន្លាស់ ព្រោះវាត្រូវបានបំប្លែងជា Machine Code ។
- Software អាច Support ការរីកចម្រើននៃ Hardware បានភាគច្រើន ។

II. Software Process :

១. • Software process : ជាសំណុំដំណើរការតាមលំដាប់លំដោយ ដើម្បីសង់ Software មួយ ដែលប្រកបដោយគុណភាព និងប្រើប្រាស់នូវពេលវេលាបានសមរម្យ ។

- Layer technology ជាសារៈសំខាន់នៃក្នុងការបង្កើត software products ។

២. នៅក្នុង Layer Technology នេះមាន Components សំខាន់ៗដូចជា :

- Quality : ជាគុណភាពរបស់ Software ដែលបំពេញបានតាម Specification គ្រប់គ្រាន់តាមប្រភេទ Product ។
- Process : ជាដំណើរការសាងសង់ Software ដែលមានការចូលរួមពី Developer និង Client requirement ដើម្បីបង្កើត Product ដែលរៀបចំ និង កែប្រែបានត្រឹមត្រូវ ។
- Method : ជាវិធីសាស្ត្ររបស់ Developer ដើម្បី build software ដែល Customer ត្រូវការទៅតាមដំណាក់កាលផ្សេងៗ ដែល Developer បានកំណត់ និង ធ្វើឡើង ។
- Tool : ជាការប្រើប្រាស់នូវ Software និង Hardware ដែលពាក់ព័ន្ធដើម្បីសាងសង់ Products ។

៣. ☛ Life cycle : ជាការបង្ហាញពីការរស់នៅចាប់ផ្តើមពីកើតឡើង, រីកលូតលាស់, រស់នៅ, និង ស្លាប់ទៅវិញ ។

☛ Software life cycle model : ជាលក្ខណៈនៃការ Develop software តាម Stage ដូចខាងក្រោម :

- Build first version of software Product ។
- Modify until Client Satisfied ។
- Operation Mode and then Retirements ។

៤. ☛ New ideal Software Life Cycle Model : ជា Model មួយ ដែលធ្វើការបង្កើតបណ្តើរ និង កែបណ្តើរដោយគ្មានការ Design រឺ Specifications ហើយធ្វើរហូតដល់ Customer មានការពេញចិត្ត ។

☛ Difficulties : ជាលក្ខណៈដែលអាចធ្វើបាន តែវាពិបាកបន្តិច ដែលមានដូចខាងក្រោម :

- បញ្ហាដែលដោះស្រាយ កាន់តែលំបាកទៅៗ ជាហេតុមិនអាចកំណត់ចំនុចបញ្ចប់បានងាយស្រួល ។
- Manager និង Developer មានការខកចិត្តចំពោះការប្រគល់ Product យឺតទៅអោយ Customer ។
- Customer មិនពេញចិត្តគុណភាព Product ដែលប្រគល់អោយ ។
- ចំនាយលុយច្រើន កែ Code ដែលពាក់ព័ន្ធនៅពេល Test និង Modification ។

☛ Problems : ជាលក្ខណៈដែលធ្វើមិនកើត ហើយព្យាយាមដោះស្រាយ :

- ការគិតមិនបានរឹងមាំ ចំពោះបញ្ហាដែលត្រូវដោះស្រាយ ។
- ខ្វះ Requirements និង យល់ច្រឡំ ចំពោះ User Specification ។
- ប្រគល់ Product ដែល Customer មិនពេញចិត្ត ។
- ចំនាយលុយច្រើនលើសគំរោង ដែលកំណត់ ។

៥. ☛ Linear Sequential Model : ជា Model មួយ ដែលធ្វើតាមលំដាប់លំដោយ ហើយទៅមុខជានិច្ច (Waterfall Model) ។

☛ Problems មានដូចខាងក្រោម :

- Project ដែលប្រើ Model នេះ ពិបាកក្នុងការផ្លាស់ប្តូរ (ងាយច្រឡំ) ។
- ប្រើ Model នេះ វាពិបាកផ្លាស់ប្តូរ requirements ដែល User បានកំណត់ ។
- Client មិនដឹងពី Version នៃការងាររហូត Project ជិតបញ្ចប់ ដែលមានបញ្ហាចំពោះការថែម Requirement ។
- Developer បាន Blocked ការងារដែលមិនសំខាន់ចោល ដែលវាមិនទាន់បានបញ្ចប់ ។

៦. ➤ The Prototyping Model ជា Model មួយ ដែលប្រើប្រាស់នូវតម្រូវការផ្នែកខ្លះ រឺ ផ្នែកថ្មី ដែលយើងជា Developer មិនយល់អំពី បំណង របស់ Customer ត្រូវការ ។ ប្រភេទនៃ The Prototyping Model មាន ២ ប្រភេទគឺ ៖

- Prototyping Model ។
- Rapid Prototyping Model ។

➤ Prototyping Model មាន Advantages ដូចខាងក្រោម ៖

- ងាយស្រួល និង លឿនលើ Client Requirements ។
- Client បានចូលរួមផ្តល់ Requirements បន្ថែម និង បំពេញ Feedback ផ្សេងៗ ។
- វាជា Model មួយដែលបំពេញលក្ខណៈដូចខាងក្រោម ៖
 - Customer អាចផ្តល់នូវ detailed requirements ។
 - ការទំនាក់ទំនងរវាង User និង System មានលក្ខណៈងាយស្រួល ។
 - Developer ងាយប្រើប្រាស់ New Technology, Hardware, និង Algorithms ។
 - Developer ងាយស្រួលកំណត់ Domain ទៅលើ Application System ។

➤ Prototyping Model មាន Problems ដូចខាងក្រោម ៖

- វាជា Model មួយដែលប្រើប្រាស់បានតែចំពោះ System ដំបូងប៉ុណ្ណោះ ។
- Developer ពឹងផ្អែកលើ Prototype សុទ្ធសាធ ដើម្បីផលិត Software Product ។
- Developer ពឹងផ្អែកលើ Prototype សុទ្ធសាធ ដើម្បីផលិត Software Product ។
- Developer ធ្វើ Implementation ទៅលើ Prototype ដែលបានកំណត់ និង យល់ព្រមយ៉ាងលឿនចំពោះការ Order របស់ Client ដោយមិនបានគិតពីភាពដែលអាចធ្វើទៅបានឡើយ ។
- អតិថិជនយល់ថា Prototype នេះវាជា Product ពិត ។

➤ Rapid Prototyping Model : វាជា very quit នៃ Prototyping Model ហើយ វាជា Model ដែលធ្វើការទៅតាម សមាមាត្រនៃសំណុំរបស់ Product ដែលមានដូចខាងក្រោម ៖

- វាធ្វើការ Interaction ជាមួយ Client បានយ៉ាងត្រឹមត្រូវទៅតាម Specification ។ វាមានលក្ខណៈស្រដៀង Linear Model ដែលអាចមាន Loop feedback តិច រឺ អត់ ។
- នៅក្នុង Specification, Planning, Design, Verification, Implementation, and Integration ត្រូវតែមាន ។
- គោលការណ៍រួមរបស់ Rapid prototype model គឺត្រូវការ Rapid នៅគ្រប់ Phase ។
- ការប្រើ Rapid prototyping គឺជាផ្តល់ការរវាង Waterfall model និង Prototype model ដើម្បីបំពេញអោយ គ្រប់នូវតម្រូវការ Client Requirements ។

៧. • RAD Model : ជា Model ដែល Develop Software ពីដំបូងលើ Components ដែលមានស្រាប់ ។

- RAD Model មាន Advantages: → ចំណាយពេលតិចនៅក្នុងការ Develop Software និង កាត់បន្ថយការចំណាយ ព្រោះគេប្រើប្រាស់ Software និង Component ដែលមានស្រាប់ ។
- RAD Model Disadvantages : → សំរាប់ Project ធំ ដែលមិនអាចពង្រីក ឬ ពង្រួមបាន ។ RAD ត្រូវការ Resource ច្រើន ។
 - RAD model តំរូវអោយ Customer និង developer ប្តេជ្ញាធ្វើតាម Schedule ។
 - ការសាងសង់ Software ជា Project specific ដែលមិនអាចបំបែកបាន ។ → គុណភាពរបស់ Software ផ្អែកទាំងស្រុងលើ Components ដែលមានស្រាប់ៗ ។ → មិន Support projects ដែលប្រើ Technique risk និង New Technology ។

៨. The Evolutionary Process Model : ជា Model មួយដែលប្រើប្រាស់តែរបស់មានស្រាប់ ។ វាច្រើនប្រើសំរាប់លក្ខណៈ System Maintenance ។ នៅក្នុង Model នេះមាន Sub model សំខាន់ៗដូចជា : Incremental Model, Spiral model, និង Win Win Spiral model ។

- Incremental model : ជាការបន្ថែម Model ដោយធ្វើការពង្រីកបន្ថែមទៀត ពោលគឺ : ធ្វើបណ្តើរ និង ចែកបណ្តើរ ។
លក្ខណៈរបស់វាមានដូចខាងក្រោម :
 - យើងបង្កើតវាដោយប្រើ Designed, Coded, and Integrated លើ Software ដោយ Test លើគ្រប់ផ្នែក ។
 - វាអាចមាន Risk នៅពេលមានការ Deliver ទៅអោយអតិថិជនជាបណ្តើរៗ ។
 - រាល់ផ្នែក ដែលក្លាយជា Product ត្រូវបានបង្ហាញអោយ Client តាមជាថ្នាក់ៗបន្តបន្ទាប់គ្នា ។
 - បញ្ហាដែលបានក្រោម និង បញ្ហាដែលអាចផ្លាស់ប្តូរនៅក្នុង Source Requirements ។
- Incremental model : មានគុណសម្បត្តិដូចជា :
 - ការបង្កើត Software មានលក្ខណៈលឿន និង ឆាប់រហ័ស នៅក្នុង Software Life Cycle ។
 - មានលក្ខណៈ Flexible ដោយចំណាយលុយតិចក្នុងការ Changes Scope Requirements ។
 - ងាយស្រួល test និង Debug ទៅលើ Smaller Iteration ។
 - ងាយស្រួលគ្រប់គ្រង Risk ព្រោះយើងបានកំណត់ និង ចាត់ចែងនូវអ្វី ដែលរកឃើញ ។
 - រាល់ Iteration នីមួយៗ មានព្រំដែន និង ទីតាំងកំណត់ជាក់លាក់ ។
- Incremental model : មានគុណវិបត្តិដូចជា :
 - នៅក្នុង Phase របស់ Iteration ពិបាកក្នុងការកែប្រែ ។
 - Problem អាចកើតឡើងនៅពេលធ្វើការតំឡើង System ព្រោះ Requirements ទាំងអស់មិនត្រូវបានបញ្ចូលគ្នានៅក្នុង Software Life Cycle ។
- Spiral Model : ជា Model ដែល Control លើ Linear Sequential Model តាមរូបសណ្ឋានជាប្រព័ន្ធរបស់ Nature Prototyping Model ។ លក្ខណៈរបស់វាមានដូចខាងក្រោម :
 - វាផលិត Software ដោយបង្កើតជា Series នៃការចែកចាយជាបន្តបន្ទាប់ ។
 - នៅក្នុងការចែកចាយលើកដំបូង វាគ្រាន់តែមានទំរង់ជាក្រដាស ហើយមិនទាន់ក្លាយជារបស់ពិតនៅឡើយ ។
 - មានបន្ថែមនូវ Framework Activities ដូចជា : Customer Communication, Planning, Risk Analysis, Engineering, Construction and Release (Setup File and Submit Software អោយ Customer) និង Customer Evaluation ។

- វាជា Model ដែល Keep នូវការ Revisiting ទៅលើ System តាមរយៈ Life time របស់វា ។

➤ Win Win Spiral Model : ជា Model មួយ ដែលធ្វើការបន្តពី Spiral Model តាម Step សំខាន់ៗមានដូចជា :

- បន្ទាប់ពីកំណត់ Framework Activity (សកម្មភាពគំរូ) រួចរាល់ វាកំណត់នូវ Level បន្ទាប់ ដែលទាក់ទងនឹង Project ដែលបានកំណត់ ។
- ពិនិត្យនូវលក្ខខណ្ឌ ដែលអាចឈ្នះ ។
- ផ្សះផ្សានូវលក្ខខណ្ឌ ដែលអាចឈ្នះ និង ត្រឹមត្រូវ ហើយបែងចែកនូវ Level បន្តដែលបានកំណត់ទុក ។
- ប៉ាន់ស្មាននូវជំរើសនៃ Product Process និង Resole risk ។
- កំណត់នូវ Level បន្ទាប់លើ Product និង Process ដែលបានចែកជាផ្នែកៗ ។
- បញ្ជាក់សារឡើងវិញនូវភាពទទួលយកបាននៃ Product and Process Definition ។
- Review និង ប្តេជ្ញាថាធ្វើវាអោយបាន ។

III. Software Metrics :

១. Metric : ជាឧបករណ៍ ដែលគេបង្កើតឡើងសំរាប់វាស់អ្វីមួយ ។

២. Defects គឺជា Error ដែលបានកើតឡើងនូវពេលដែល Software ក្លាយជា Real Products ។ Defects មាន ៣ ប្រភេទគឺ :

- Severe Defect : កើតឡើងទាក់ទងទៅនឹង Functionality នៃ Product ។
- Major Defect : កើតឡើងទាក់ទងទៅនឹង Logicality នៃ Product ។
- Minor Defect : កើតឡើងទាក់ទងទៅនឹងតំរូវការរបស់ Client (Data Entry) ទៅលើភាពចូលចិត្ត និង មិនចូលចិត្ត ។

៣. Characteristics of Metric : នៅពេលគេបង្កើត Metric វាមានលក្ខណៈដូចខាងក្រោម :

- Goal Oriented Approach : មានគោលដៅមួយច្បាស់លាស់ ។
- Measurable : វាស់បានទៅលើអ្វីដែលវាវាស់បាន ។
- Analyzable : អាចវិភាគបាន ដើម្បីទទួលយកការកែប្រែណាមួយ ។
- Programming Language Independent : វាវាស់ទៅតាមប្រភេទ Program ផ្សេងៗគ្នា ដែលមិនទាក់ទងគ្នា ។
- Timely : ប្រើប្រាស់នូវពេលវេលាគ្រប់គ្រាន់ក្នុងការបង្កើត និង កែប្រែ ។

៤. Step to Create Metric : ដើម្បីបង្កើត Metric គេប្រើប្រាស់ Step ដូចខាងក្រោម :

- កំណត់ពីគោលបំណង នៃការបង្កើត Metric ។
- កំណត់នូវតំរូវការរបស់ Metric ។
- កំណត់នូវតំលៃជឿជាក់មួយ (Base Line Value) ទៅអោយ Metric ។
- ពិនិត្យនូវភាពប្រើប្រាស់បានទៅតាមពេលកំណត់ណាមួយ ។

៥. Type of Software Metrics មានដូចខាងក្រោម :

- Design Metric : ជា Metric សំរាប់វាស់នៅលក្ខណៈ Design time ដោយពិនិត្យលើ Architecture design metric ។
លក្ខណៈរបស់ Architecture design metric មាន :
 - Structural Complexity : កើតឡើងនៅពេលចំនួន Fan-out module ត្រូវបានហៅដោយ Parents ច្រើនដង ។
 - Data Complexity : កើតឡើងនៅពេលមានការវិភាគលើ data ផ្នែក Input និង Output ច្រើនដង ។
 - System Complexity : កើតឡើងអាស្រ័យលើទំហំនៃ Structure និង Data Complexity របស់ System ។
- Project Metric : ជា Metric ដែលគេបង្កើតឡើងសំរាប់វាស់ទៅលើ Project ជាក់ស្តែង ។ វាមានផ្នែក Metric សំខាន់ៗដូចជា :
Effort, Productivity in FP, Cost, Size, Defects, និង Testing ។
- វាមានលក្ខណៈដូចខាងក្រោម :
 - Effort metric : Metric សំរាប់វាស់រយៈពេលធ្វើ Project សុទ្ធសាធ ដោយដកថ្លៃសំរាកចេញ ។
 - Productivity metric : Metric សំរាប់វាស់លទ្ធភាព រឿនរបស់ Product ដែលត្រូវផលិត ។
 - Cost metric : Metric សំរាប់គណនាបរិមាណទឹកប្រាក់ចំណាយសរុបសំរាប់ Hold Project ។
 - Size metric : Metric សំរាប់វាស់ Size របស់ Project គិតជា FP (Function Point) ។
 - Defects metric : Metric សំរាប់វាស់ mistake របស់ Real Product ពី Client Report ។
 - Testing metric : Metric សំរាប់វាស់លើចំនួនករណីនៃការ Test ទៅលើ Software ដែលមានផ្នែកពីរសំខាន់ៗគឺ :
Integration testing (Test គ្រប់ផ្នែកបញ្ចូលគ្នា) និង Unit testing (Test ដោយផ្នែកៗ) ។
- Product Metric : ជា Metric សំរាប់វាស់គុណភាពរបស់ Software Product រវាងកំរិតគុណភាពជាក់ស្តែង និង
គំរោងដែលបានកំណត់ទុកនៅក្នុង Plan ។
- Maintenance Metric : ជា Metric សំរាប់ maintenance ទៅលើ Project ដែលបានបង្កើតឡើង ។
- Maintenance metric គេ Focus ទៅលើលក្ខណៈមួយចំនួនដូចជា :
 - ប្រភេទនៃការ Change វាមានដូចជា : Correction, Enhancement, Adaptation, Re-engineering ។
 - ការផ្លាស់ប្តូរ Trigger Component វាមានដូចជា : Requirement, Design, Code, និង Other ។
- ប្រភេទនៃការប៉ាន់ស្មានទៅលើដំណើរការ maintenance ទៅលើ Project មាន ២ ប្រភេទគឺ :
 - ការបំពេញលក្ខណៈនៃការ Change Requirement ដោយ Developer ។
 - ការបំពេញលក្ខណៈនៃការ Request ពី Customer ។

IV. Measuring effort for software Project :

- **SLOC Technique** (Source Line of Code Technique) : វាជា Technique សំរាប់វាយតម្លៃ វី ប៉ាន់ស្មាន Size នៃ Project ។ ដែល Size របស់ Project វាអាស្រ័យទៅលើលក្ខខណ្ឌមួយចំនួនដូចជា : Resource, Effort, Cost, និង Duration ដើម្បីធ្វើ Project នោះបានសម្រេចឡើង ។
- វាអាចវាស់តម្លៃដោយផ្ទាល់ទៅលើ Effort ដែលត្រូវចំនាយលើទៅលើ Project តែម្តង ។
- SLOC ដែលត្រូវបានរាប់មាន :
 - Line of code ដែលទទួលពី Client ។
 - Code ដែលសរសេរផ្ទាល់ដោយ Developer ។
 - The decoration ។
- SLOC ដែលមិនត្រូវបានរាប់មាន :
 - Comment របស់ Developer ។
 - Code ដែល Generate ដោយ Application generator ។
- Technique សំរាប់រក Effort មាន : **SLOC Technique** (Source Line of Code Technique) , **FP Technique** (Function Point Technique), **COCOMO Technique** (Constructive Cost Model Technique) and **Delphi Technique**.
- កំណត់សំគាល់ :
 - វាអាចវាស់បានអាស្រ័យលើ Language ដែលយើងបានកំណត់ វី មានបទពិសោធន៍ទៅលើវា ។
 - វាធ្វើបានល្អប្រសើរត្រឹមត្រូវ Language ដើម និង ថ្មីមានលក្ខណៈខុសគ្នា ។
- Delphi technique គឺជា technique មួយ ដោយពឹងផ្អែកលើការ Estimate របស់មនុស្សសុទ្ធសាធ ។
- Delphi technique ប្រើប្រាស់ technique ដោយយក Experience និង បំនិននៅក្នុងការ Analyst ដើម្បី Estimate ទៅលើ Size, productivity, និង Effort ដែលជាតំរូវការរបស់ Project ។
- Delphi technique មាន step ចំនួន ៨ គឺ :
 - **Step 1 :** វិភាគអ្នកចូលរួម ដើម្បីដំណើរការសកម្មភាព Estimation ។
 - អញ្ជើញអ្នក Expert ជា Group ដែលមាន Experience លើ Project ចាស់ និង មានលក្ខណៈសមរម្យសំរាប់ Project ដែលកំពុងធ្វើមក Estimate ។
 - អញ្ជើញអ្នកសម្របសម្រួល (Project coordinator) មកសម្របសម្រួល ដើម្បីបង្ហាញអោយឃើញថា Meeting ដែលបង្កើតឡើងមានភាពជឿជាក់ និង ទទួលបានលទ្ធផលដែលទទួលយកបាន ។
 - អញ្ជើញ Author (Recorder) ដើម្បីកត់ត្រាការងារ ដែលបានធ្វើ ។
 - **Step 2 :** បង្ហាញ Project detail ទៅ Group Expert ។
 - Author បង្ហាញ Project detail ដែលមានតំរូវការរបស់ Customer ទៅអោយ Group expert ។
 - Author បង្ហាញការសង្ស័យមុខកំពោះ Group expert ។
 - Author និង Expert ធ្វើការ analyze ទៅលើ Task ដែលត្រូវ Estimate ហើយ សន្និដ្ឋានទៅលើតំរូវការដែលពួកគេពិចារណា ដែលអាចទទួលយកបាន ។

- **Step 3 :** កំណត់តម្លៃ Acceptable variance value ។
 - Expert និង Author ទទួលបានមតិរួមដែលធ្វើការ Estimate ដើម្បីទទួលយកតម្លៃ ដែលមិនទទួលយកបាន ។
- **Step 4 :** រៀបចំ Task ដែលត្រូវធ្វើ ។
 - Project coordinator ភ្ជាប់ Task list ដែលបែងចែកដោយ team ទៅអោយ Expert ។
- **Step 5 :** ការ Estimate ដែលសំរេចដោយ Expert ។
 - Expert ធ្វើការ Estimate លើការងារនីមួយៗ ហើយ Note ដោយដៃ បន្ទាប់មកបញ្ជូនអោយ Project Coordinator ។
 - កំលុងពេល Estimate គ្មានការពិភាក្សា រឺ ផ្តល់យោបល់ ពីព្រោះការ Estimate logic អាស្រ័យលើ Logic របស់ Expert ជាអ្នកកំណត់ ។
 - ចំនែក coordinator និង author ត្រូវជឿជាក់លើលទ្ធផលទាំងនេះ ។
- **Step 6 :** រៀបចំដោយសង្ខេបលើការបូកសរុបលើ Task នីមួយៗ ។
 - Coordinator រៀបចំការ Estimate នីមួយៗ ហើយរក % variance ដែល Valid និង Invalid ។
- **Step 7 :** ពិភាក្សាលើ Task ដែលមិនទាន់ Valid Result ។
 - Coordinator ប្រគល់ការងារដែល invalid អោយ Expert និង author ធ្វើការពិភាក្សាលើ Task នោះ រហូតទទួលបានតម្លៃជឿជាក់ណាមួយ ដែលអាចយកទៅប្រើប្រាស់បាន ។
- **Step 8 :** ធ្វើការងារនៅ Step ទី ៥ រហូតដល់ការ Estimation ត្រូវបានបញ្ចប់ ។

V. Analysis modeling :

១. Element Analysis Model : នៅក្នុង Analysis Modeling គេផ្តោតលើលទ្ធផល ៣ ប្រភេទសំខាន់ៗគឺ :

- អធិប្បាយអំពីតម្រូវការរបស់ Customer ។
- ចាប់ផ្តើមធ្វើនូវការបង្កើតឡើងនៃ Software design ។
- កំណត់នូវសំនុំ Requirements ដែលត្រឹមត្រូវមួយសំរាប់ Build Software នោះឡើង ។

២. Data Modeling ជា Model ដែលឆ្លើយតបនូវសំនុំសំនួរដែលពាក់ព័ន្ធ ទៅលើ Data Processing នៅក្នុង Application ។

នៅក្នុងនោះ គេបានប្រើ ERD (Entity Relationship Diagram) ដើមកំណត់ Data ដែលត្រូវបញ្ចូល រក្សាទុក ដំនើរការ ដែលធ្វើឡើងដោយ Application ផ្ទាល់ ។ ERD វាបានបង្ហាញអំពីផងដែរនូវ data និង Relationship ដើម្បីអោយ Application មួយនេះអាច Process បានល្អ ។

៣. ➤ Data Object : គឺជា Information ដែលប្រើសំរាប់តំណាង និង បង្កើតឡើងដើម្បីអោយ Software អាចយល់បាន ។

➤ Attribute : គឺជា Information ដែលរៀបរាប់អំពីលក្ខណៈរបស់ Object មួយ ។ គេប្រើវា :

- កំណត់ឈ្មោះទៅអោយ Data Object ណាមួយ ។
- ពណ៌នាលក្ខណៈអោយ Data Object នោះ ។
- ធ្វើ Reference ទៅកាន់ Data Object ណាមួយទៀត ដែលមាននៅក្នុង Table ។

➤ Relationship : គឺជាទំនាក់ទំនងដែលអាចកើតមាននៅពេលមាន Data Object ពីរ ឬ ច្រើន Connected រវាងគ្នានឹងគ្នា ។
Relationship ទាំងនោះរួមមានដូចជា (Bookstore and Book):

៤. ➤ Cardinality : ជាចំនួន Object ដែលមានទំនាក់ទំនងគ្នាទៅវិញទៅមក ដោយមិនអាចខ្វះបាន ។

➤ Modality : ជាចំនួន Object ដែលមានទំនាក់ទំនងគ្នាទៅវិញទៅមក ដោយអាចករណីលើកលែងទៅតាម Object ដែលវាធ្វើការទំនាក់ទំនងគ្នា ។

➤ Relationship ទាំងនោះរួមមាន :

- One-to-One (1:1): A មានទំនាក់ទំនងត្រឹមតែមួយនឹង B ហើយ B មានទំនាក់ទំនងត្រឹមតែមួយនឹង A ។
- One-to-Many (1:N): A មានទំនាក់ទំនងច្រើននឹង B ហើយ B មានទំនាក់ទំនងត្រឹមតែមួយនឹង A ។
- Many-to-Many (M:N): A មានទំនាក់ទំនងច្រើននឹង B ហើយ B មានទំនាក់ទំនងច្រើននឹង A ។

៥. ➤ Functional Modeling and Information Flow :

• Functional Modeling : ជាចំនួន Function ដែលដំនើរការនៅក្នុងផ្នែកណាមួយរបស់ System Application ។ វាប្រើសំរាប់ Apply លើ Hardware, Software, Human elements, និង Produce នូវ Output ។

• Information Flow : ជាបរិមាណ ឬ ចំនួន Data ដែលត្រូវបាន Analysis ដោយ System ដែលបាន ទទួល រឺ Transformed ពី External Entity ។

➤ Behavioral Modeling : ជាដំណើរការវិភាគទៅលើការ Analysis Requirements ទៅតាម Version របស់ Structure ដែល Provide ដោយ Node នីមួយៗ របស់ Model ។