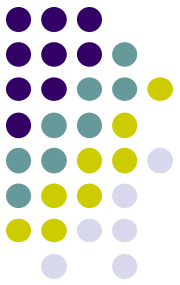


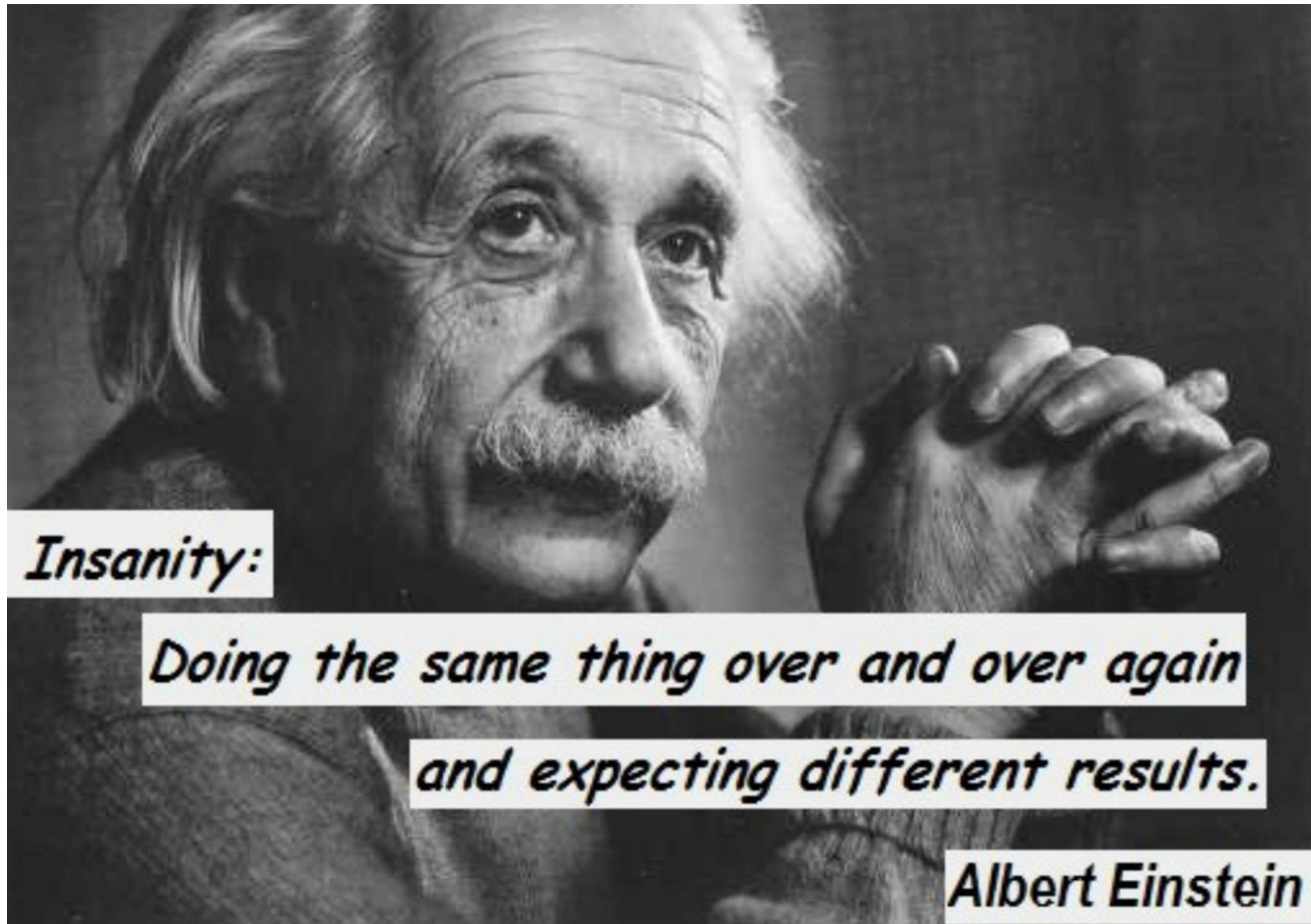
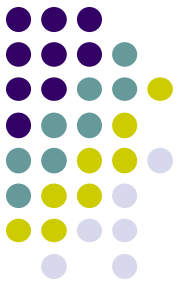
MSc. Pok Leakmony (ប៉ុក លក្ស្មុនី)

Lecturer of CSD, RUPP



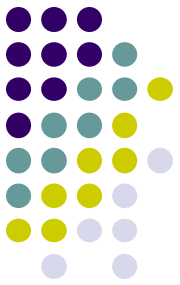
Teaching Subjects:

- **Software Engineering (Bachelor class, Y4)**
- **IT Project Management (Bachelor class, Y4)**
- **.NET Programming (Bachelor class, Y3)**
- **C Programming (Bachelor class, Y1)**
- **Distributed Database Systems (Master class)**



Semester 1

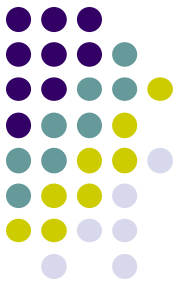
Software Engineering(SE)



- **Chapter 1: Introduction to SE ([1], [2])**
- **Chapter 2: The Process ([1], [3], [6], [7])**
- **Chapter 3: Software Project Management Metrics ([1], [4])**
- **Chapter 4: Measuring Effort for Software Project ([1], [4], [6], [9])**
- **Chapter 5: Analysis Concepts and Principles ([1], [7])**
- **Chapter 6: Analysis Modeling ([1], [7])**
- **Chapter 7: Software Testing Techniques and Strategies ([1], [5], [7], [8])**

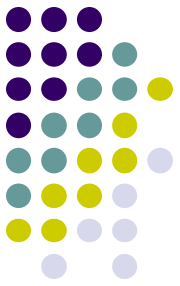
Semester 1

Software Engineering(SE)



Reference Text Books

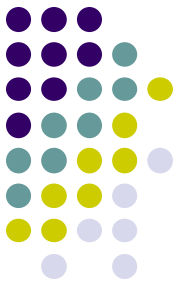
- [1] Software Engineering: A Practitioner's Approach, Roger.S.Pressman, Fifth Edition 2001
- [2] Object-Oriented Software Engineering: Practical software development using UML and Java, Timothy C. Lethbridge and Robert Laganière, International Edition, 2002
- [3] Software Engineering, Ian Sommerville, 6th Edition, 2000
- [4] Basic of Software Project Management, NIIT, Eastern Economy Edition, 2004
- [5] Software Testing Techniques, Boris Beizer, 2nd Edition, 2003
- [6] Software Project Management, Bob Hughes and Mike Cotterell, 2001
- [7] Software System Development: A Gentle Introduction, Carol Britton & Jill Doake, Third Edition, 2003
- [8] Software Testing as a Service, Ashfaque Ahmed, 2010 (E-book)
- [9] Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, Capers Jones, 2010 (E-book)



Chapter 1

Introduction to SE ([1], [2])

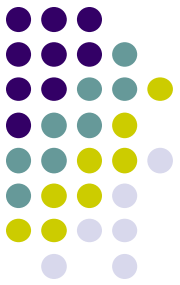
1.1 The History of SE



Some of the reasons to which you can attribute the software crisis include:

- Software developers used multiple programming languages
- Software developers used multiple variations of standard programming languages
- Most of the requirements were complex with regard to the existing capabilities
- Users, who had little or no experience of developing or even using software, formulated requirements

1.1.The History of SE

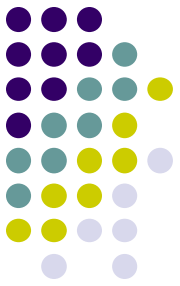


- Software developers poorly mapped requirements to the actual product
- Software developed had low interoperability
- Software maintenance was costly
- Hardware developed at a faster rate than software(better hardware requires better software to operate it)

1.1.The History of SE

- ជារួម នៅទសវត្សទី៦០ បណ្តាប្រទេសរីកចម្រើន ជួបនូវស្ថានភាពរួមមួយ ដែលគេចាត់ទុកថាជា វិបត្តិ SW (SW Crisis) គឺ: 70% នៃគម្រោងពាក់ ព័ន្ធនឹងព័ត៌មានវិទ្យាត្រូវបរាជ័យ។
- NATO បានរៀបចំសន្និសីទវិទ្យាសាស្ត្រនៅទីក្រុង Munich (1968) ដើម្បីដោះស្រាយវិបត្តិទាំងនោះ។
- \Rightarrow កើតចេញជា Software Engineering (SE) នេះឡើង។

Why SE?



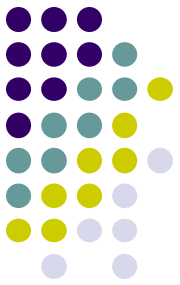
- Build it correct the first time
- Reduce development cost
- Reduce development time
- Repeat previous successes
- Learn from previous failures

Goal of SE



- To improve the quality of the software products.
- To increase the productivity &
- To give job satisfaction to the software engineers.

1.2. Concept



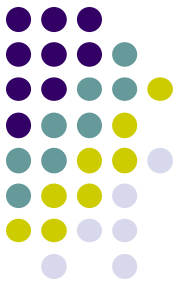
a) What is Software?

- Software is:

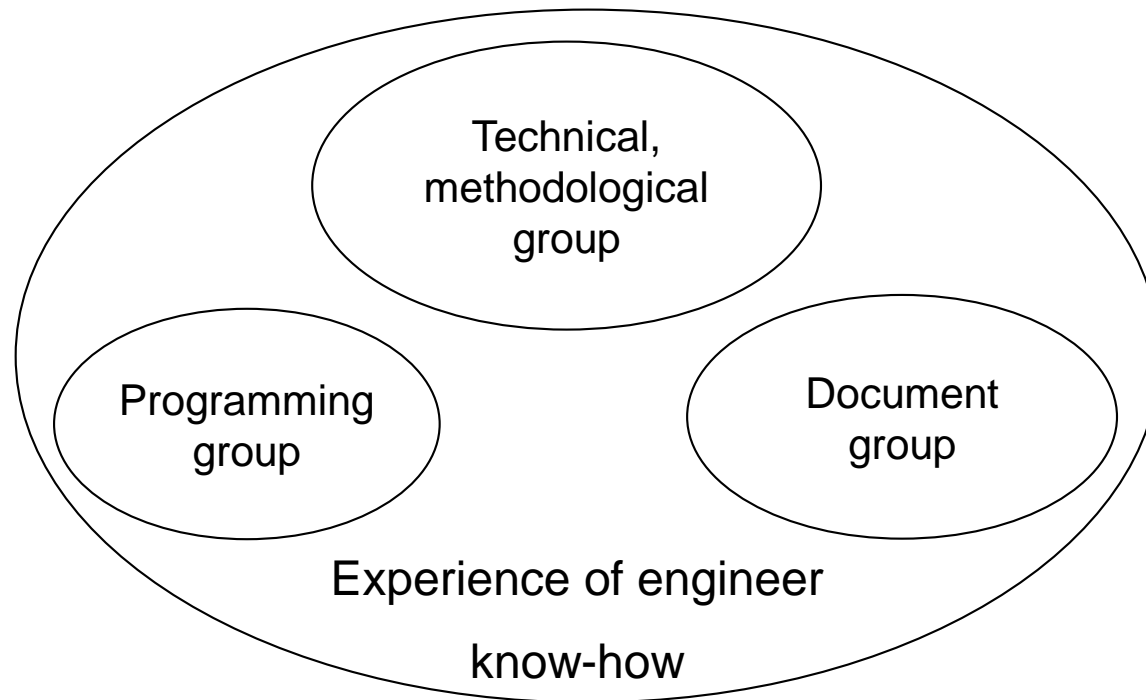
- Instructions (computer programs) that when executed provide desired function and performance.
- Data structure that enable the programs to adequately manipulate information
- Documents that describe the operation and use of the programs

* **Software:** is a set of instructions, written by one or more programming language to perform automatically some functions in solving any problem.

1.2. Concept



a) What is Software?



1.2. Concept



b) What is Software Engineering?

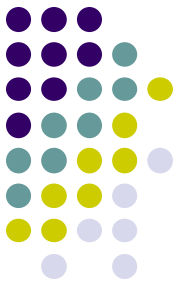
- **Engineering:** គឺជារបៀបប្រើប្រាស់នូវរាល់ឧបករណ៍ បច្ចេកទេសក្នុងរបៀបដោះស្រាយបញ្ហាណាមួយ។

- **Engineering** is the systematic application of scientific knowledge in creating and building cost-effective solutions to practical problems in the service of mankind.

- **Engineering** in general employs the following sequence of steps:

- Problem identification
- Analysis of the problem
- A search for possible alternative solutions
- Evaluation and then choice of the most appropriate solution
- Design and implementation of the solution

1.2. Concept



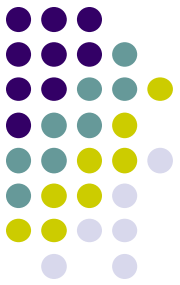
b) What is Software Engineering?

- **Software Engineering:**

- * The term Software Engineering was introduced in the late 1960s at a conference held to discuss the software crisis.
- An original definition:

“ Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines”, Fritz Bauer, seminal conference of SE, 1968

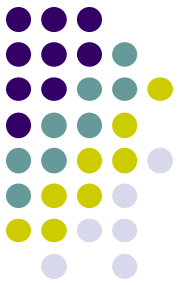
1.2. Concept



b) What is Software Engineering?

- “ SE: (1) *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.* (2) *The study of approaches as in (1).”, IEEE[1993]*
- “*SE is the process of solving customer's problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints”, Timothy C. Lethbridge and Robert Laganière[2001]*

1.2. Concept

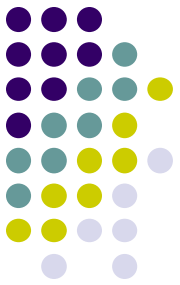


b) What is Software Engineering?

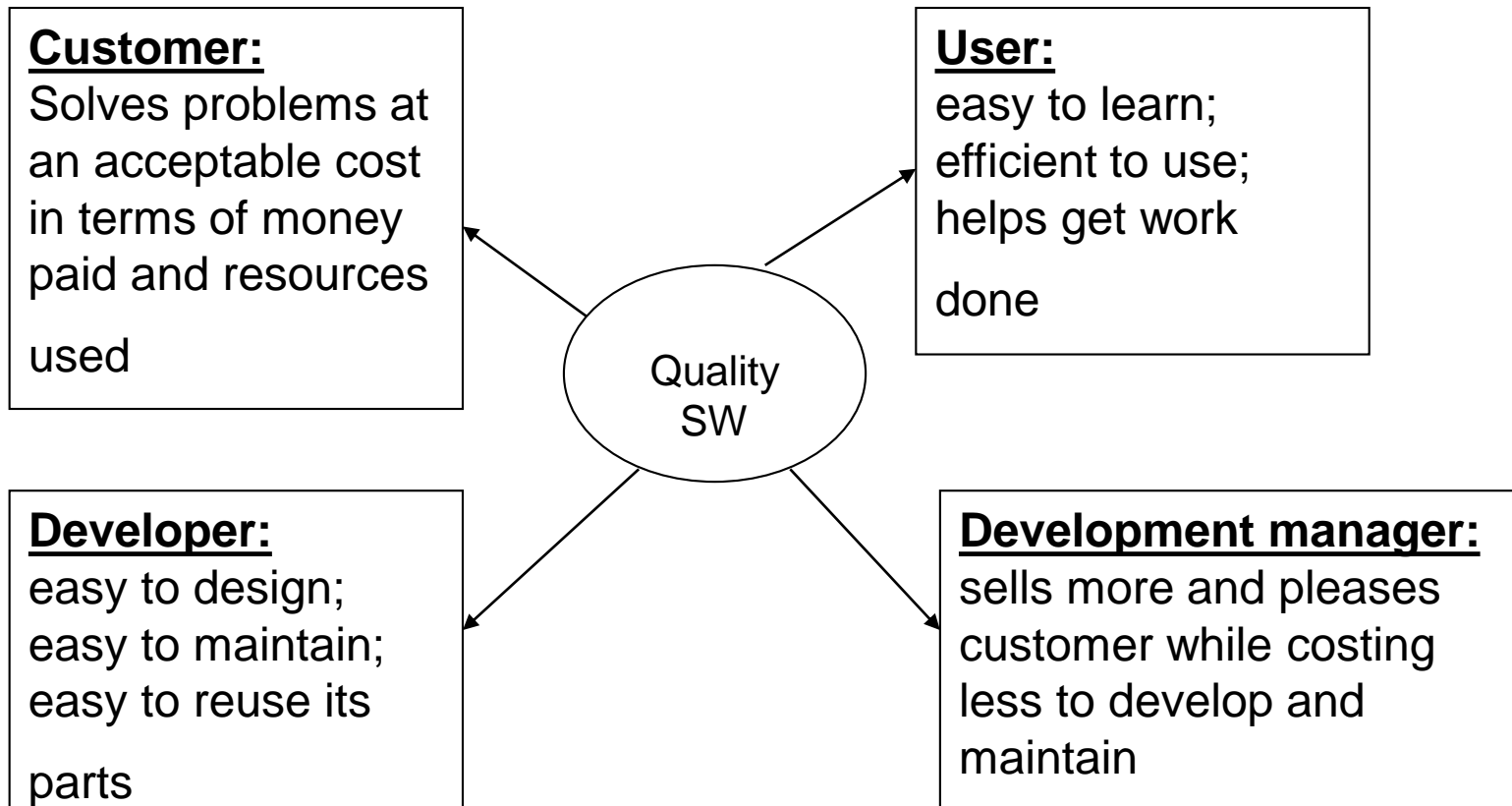
- Solving customer's problem
- Systematic development and evolution
- Large, high-quality SW system
- Cost, time, and constraints

* “Software Engineering is a scientific field to deal with methodologies, techniques and tools integrated in software production-maintenance process to obtain software with desired qualities”.

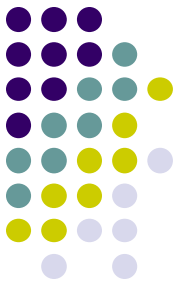
1.2.Concept



c) SW Quality



1.2. Concept

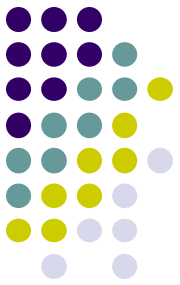


c) SW Quality

* Attributes of SW Quality

- Correctness(→): the degree to which a program operates according to specification
- Maintainability(→): the degree to which a program is amenable to change
- Integrity (→): the degree to which a program is impervious to outside attack
- Usability: the degree to which a program is easy to use
- Efficiency: the degree to which a program uses less CPU-time, memory, disk space, network bandwidth and other resources
- Reliability: the degree to which a program has a fewer failures
- Reusability: the degree to which a SW component is reusable if it can be used in several different system with little or no modification

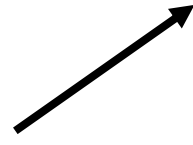
1.2. Concept



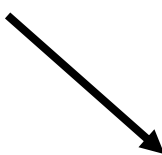
c) SW Quality

* Attributes of SW Quality

- Usability

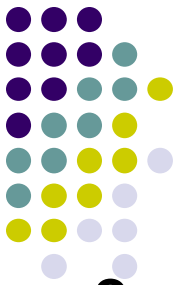


- Efficiency



Conflict (ប្រឆាំងគ្នា)

1.2. Concept



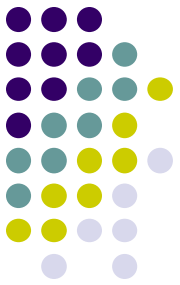
d) What is the difference between SE and computer science?

- CS is concerned with theory and fundamentals
- SE is concerned with the practicalities of the developing and delivering useful software (Ian Sommerville).

e) What is the difference between SE and system engineering?

- System engineering is concerned with all aspects of computer-based systems development, including hardware, software and process engineering.
- SE is part of this process (Ian Sommerville).

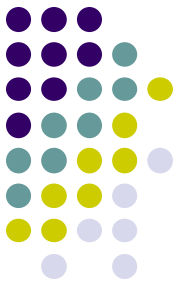
1.2.Concept



f) What is a SW Process? (Ian Sommerville)

- SW Process is a set of activities whose goal is the development or evolution of software. These activities are:
 - *SW Specification*: The functionality of the software and constraints on its operation must be defined.
 - *SW Development*: The software to meet the specification must be produced.
 - *SW Validation*: The software must be validated to ensure that it does what the customer wants.
 - *SW Evolution*: The software must evolve to meet changing customer needs.

1.2. Concept



g) SW Characteristics

- Software is developed or engineered, it is not manufactured in the classical sense.
- Most software is custom-built, rather than being assembled from existing components
- SW គឺជាទំនិញដែលគ្មានរូប មើលមិនឃើញ
- SW doesn't "wear out": ពុំមានសំណឹក។ គុណភាពរបស់វាមាននិន្នាការល្អឡើង ក្រោយពីមានកំហុសម្តងៗត្រូវបានរកឃើញ និងកែតម្រូវ។
- SW តែងមានកំហុសបង្កប់នៅក្នុងនោះតាមទ្រង់ទ្រាយកាន់តែធំលទ្ធភាពផ្ទុកកំហុសក៏កាន់តែធំដែរ។

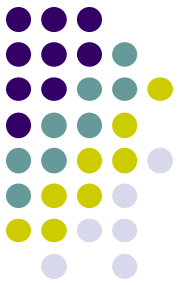
1.2. Concept



g) SW Characteristics

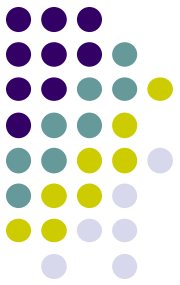
- កំហុសនៃ SW ងាយប្រទះឃើញដោយអ្នកមកពីខាងក្រៅ។
- SW តែងតែផ្ទុកដោយគំនិត និងការផ្តោតប្រឌិតនៃអ្នកសរសេរ ឬក្រុមបង្កើតវា។
- អាចធ្វើការ Copy ដោយសាមញ្ញបំផុត។

1.2. Concept



g) SW Characteristics

SW	HW
<ul style="list-style-type: none">- និយាយពីបញ្ហា “ទន់”- បច្ចេកទេសប្រើប្រាស់- មិនមែនរូបធាតុ- គ្មានរូប មិនអាចចាប់បាន- ផលិតដោយមនុស្សជាសំខាន់- គ្មានសំណឹក	<ul style="list-style-type: none">- និយាយពីបញ្ហា “រឹង” (វត្ថុ)- លោហធាតុ- រូបធាតុ- មានរូប អាចចាប់បាន- ផលិតជាលក្ខណៈឧស្សាហកម្ម ដោយគ្រឿងម៉ាស៊ីនជាសំខាន់;- ខូចបង់ មានសំណឹក



1.3. Software Applications

គេអាចចែក SW application នាបច្ចុប្បន្នចេញជា៨ ប្រភេទ:

- 1) System Software
- 2) Real-Time Software
- 3) Business Software
- 4) Engineering and Scientific Software
- 5) Embedded Software
- 6) Personal Computer Software
- 7) Artificial Intelligence Software
- 8) Web-based Software

(Homework 1)

Correctness (←)



តើគេបានអ្វីធ្វើជារង្វាស់?

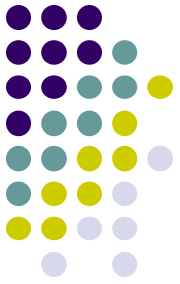
ខ្ពស់ (អត់ល្អ)

ចំនួន Defects/KLOC

ទាប (ល្អ)

- KLOC: Kilo Line of Codes
- Error ជាកំហុសកើតឡើងមុនពេល SW យកទៅប្រើ
- Defect ជាកំហុសកើតឡើងក្រោយពេល SW យកទៅប្រើ

Maintainability (←)



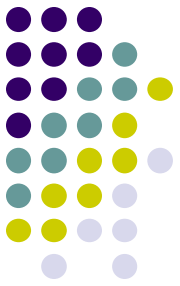
តើគេបានអ្វីធ្វើជារង្វាស់?

ខ្ពស់(អត់ល្អ)

MTTC: Mean-Time To Change

ទាប(ល្អ)

Integrity (←)



តើការវាយលុកពីខាងក្រៅចូល SW តាម
រយៈអ្វីខ្លះ?

- Virus
- Hacker