

CSE307: Software Engineering

Professional Ethics and Risks & Participants in a Project

What to Expect

- The point of studying these topics in this course is to make ourselves familiar with these concepts, so that we can relate to some of the stuffs that we will learn here later in our professional life.

Professional Ethics

The Need for Ethics

“With great power comes great responsibility.”

- Uncle Ben to Peter Parker

The Need for Ethics



Software Engineering Code of Ethics

- Software Engineering Code of Ethics was developed by the IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices (SEEPP).
- This Code was mainly developed with a view to guiding software engineers in making moral and ethical decisions while working in a software company.

Software Engineering Code of Ethics

In accordance with their commitment to the health, safety, and welfare of the public, software engineers shall adhere to the following 8 (eight) principles.

1. **Public**: Software engineers shall act consistently with the public interest.

2. **Client and Employer**: Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

Software Engineering Code of Ethics

In accordance with their commitment to the health, safety, and welfare of the public, software engineers shall adhere to the following 8 (eight) principles.

3. Product: Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. Judgement: Software engineers shall maintain integrity and independence in their professional judgement.

Software Engineering Code of Ethics

In accordance with their commitment to the health, safety, and welfare of the public, software engineers shall adhere to the following 8 (eight) principles.

5. Management: Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. Profession: Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

Software Engineering Code of Ethics

In accordance with their commitment to the health, safety, and welfare of the public, software engineers shall adhere to the following 8 (eight) principles.

7. Colleagues: Software engineers shall be fair to and supportive of their colleagues.

8. Self: Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Software Engineering Code of Ethics

- However, this Code is not a simple ethical algorithm which generates ethical decisions.
- Based on the principles, you will assess the options available and make the best judgement out of them.

Software Engineering Code of Ethics

- You will find the full version of the Code of Ethics [here](#).

Risks in a Project

Risk Management

As a participant in a software project, you need to find the answers to the following questions while evaluating the risks in your project.

1. What can go wrong?
2. What can the potential damage be?
3. What can we do about it?

Risk Identification

- A risk is a potential problem which may or may not happen in the future. (**Uncertainty**)
- If it happens, then it brings about some unwanted consequences. (**Loss**)

Risk Identification

Risks in a project can be categorized as follows based on their effect.

1. **Project Risk:** It threatens the project plan. If it becomes real, then it may delay the project schedule resulting in the increment of the project cost.
2. **Technical Risk:** It threatens the quality and timeliness of the software product to be produced. If it becomes real, then it may complicate the implementation of the product.
3. **Business Risk:** It threatens the viability of the software product to be produced. If it becomes real, then it may jeopardize the project.

Risk Identification

Risks in a project can be categorized as follows based on their nature.

1. **Known Risk:** It can be identified after careful evaluation of different aspects of the project. Unrealistic delivery date of a software product may fall into this category.
2. **Predictable Risk:** It can be extrapolated from the past project experience. Previous employee turnover may fall into this category.
3. **Unpredictable Risk:** It is extremely difficult to identify or estimate in advance.

Risk Estimation

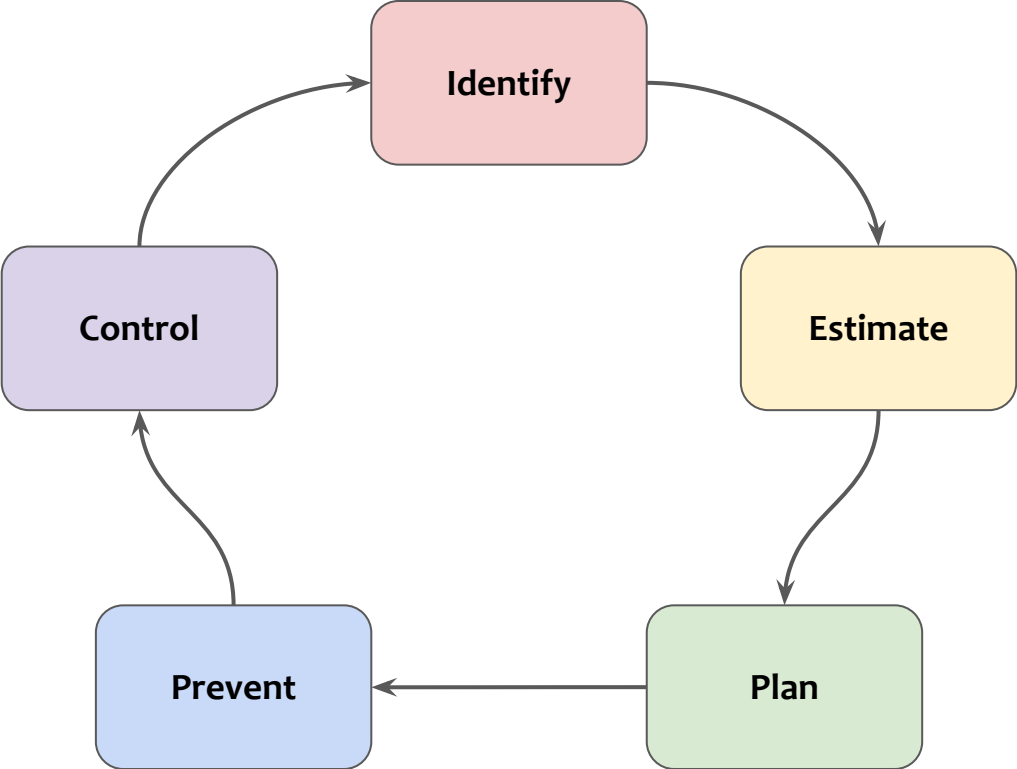
- Risk estimation, also known as risk projection, attempts to rate each risk in 2 (two) ways.
 1. It attempts to estimate the likelihood of a risk.
 2. It attempts to determine the consequence of the problems associated with a risk.
- The risks associated with a project are, then, ranked based on their probability and severity so that they can be addressed on a priority basis.

Risk Prevention

There are 2 (two) strategies for dealing with risks in a project.

1. **Reactive Risk Strategy:** Nothing is done about risks until something actually goes wrong. The project team then rushes to take necessary actions in an attempt to correct the problem quickly (firefighting).
2. **Proactive Risk Strategy:** Objective is to avoid risks and to device contingency plan to handle unavoidable risks in a effective manner (disaster prevention).

Risk Management Paradigm



Risk Table

Risk	Category	Probability	Impact	RMMM
Size estimate may be significantly low	PS	60%	2	
Larger number of users than planned	PS	30%	3	
Less reuse than planned	PS	70%	2	
End users resist system	BU	40%	3	
Delivery deadline will be tightened	BU	50%	2	
Funding will be lost	CU	40%	1	
Customer will change requirements	PS	80%	2	
Technology will not meet exceptions	TR	30%	1	
Lack of training on tools	DE	80%	3	
Staff inexperienced	ST	30%	2	
Staff turnover will be high	ST	60%	2	

Impact values:
1 – catastrophic
2 – critical
3 – marginal
4 – negligible

Risk Information Sheet

Risk information sheet			
Risk ID: P02-4-32	Date: 5 / 9 / 19	Prob: 80%	Impact: high
Description: Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.			
Refinement/context: Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards. Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components. Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.			
Mitigation/monitoring: 1. Contact third party to determine conformance with design standards. 2. Press for interface standards completion; consider component structure when deciding on interface protocol. 3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.			
Management/contingency plan/trigger: RE computed to be \$20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. Trigger: Mitigation steps unproductive as of 7 / 1 / 19.			
Current status: 5 / 12 / 19: Mitigation steps initiated.			
Originator: D. Gagne		Assigned: B. Laster	

Participants in a Project

4 P's of Project Management

A software project can be characterized in terms of its following 4 (four) components.

1. **People**: The individuals who contribute to a software project in some way.
2. **Product**: The software to be built for the client.
3. **Process**: The set of software engineering activities to get the job done.
4. **Project**: All efforts required to make the product a success.

Participants in a Software Project

Different participants (or stakeholders) play different roles in a software project.

- **Practitioner:** They provide technical knowledge and skills essential for engineering a software product.
- **Project Manager:** They plan and organize different aspects of a software project, communicate with the client and the upper management for business related issues, motivate and supervise the practitioners to make the project a success.
- **Client:** They specify the requirements for the software product and interact with the software after the release.

Participants in a Software Project



Software Team Selection

There are some factors to consider while selecting a team for a software project.

- Complexity of the requirements provided for the software product.
- Degree to which the tasks can be modularized effectively.
- Quality and reliability of the product to be built.
- Rigidity of project deadline.
- Amount of communication required for the project.

Management vs Leadership

- Management means controlling a set of individuals to accomplish a goal.
- Leadership means inspiring a set of individuals to accomplish a goal.
- Both management with poor leadership and leadership with poor management tend to cause the failure of a software project.