



**MCI**  
MANAGEMENT CENTER  
INNSBRUCK



## SYSTEMS PLANNING


### Software Project Planning

DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu

1




**MCI**  
MANAGEMENT CENTER  
INNSBRUCK

## Class content

### What to expect from this class?

- A discussion of factors influencing **project planning and scheduling**
- A **comparison of plan-driven and agile** software project planning
- An introduction to **software development cost estimation techniques**




DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu


2

## Class learning goals



### What should be your learning outcome?

- Understand the **factors influencing the development schedule** of software
- Know the **difference between plan-driven and agile software project planning**
- Know **different software cost estimation techniques**



Source: <https://pixabay.com/en/learning-hint-school-subject-3245792/>  
by harishs | CC0 Creative Commons


DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu

3

## Software Project Planning




### Software Project Planning

- One of the **most important jobs** of a software project manager
- **Break down** the work into parts, **assign** parts to team members, **anticipate** problems, **prepare** solutions

**Project planning takes place**

- During the proposal stage
- During the project startup phase
- Periodical, throughout the project



Source: <https://pixabay.com/en/project-management-business-planning-2061835/>  
by Tumen | CC0 Creative Commons


DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu

4

## Software Project Planning




### Project costs

- Planning at the **proposal stage is speculative**
- It is **based on high-level descriptions**
- You need to **estimate the cost by working out how much effort is required**

There are parameters you should use when computing costs:

- **Effort** cost (the cost of paying engineers and managers)
- **Hardware and software** costs (incl. maintenance)
- **Travel and training** costs



Source: <https://pixabay.com/en/coins-banknotes-money-currency-1735615/>  
by stephebl / CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

5


## Software Project Planning



### Project costs

- For most projects the **effort costs are the biggest**
- You have to estimate the **effort in person month**
- **Software costs** can also be significant (e.g. licenses for **middleware and platform software**)
- Extensive **travel** may be needed when the project is developed at **different sites** (you may use electronic meeting software to reduce costs)

**Note:** Once the contract to create a project has been awarded the outline project plan has to be **refined**. The resulting **project plan** can be used to support decisions about **project staffing or budgeting**.



Source: <https://pixabay.com/en/application-money-1756279/>  
by koutre / CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)


6

## Software Project Planning



### Software Pricing

- **In principle** the price of a software product is the **cost of development plus profit for the developer(s)**
- **In practice** this calculation is usually **not so simple**
- Broader **organizational, economic, political and business consideration** may cause a price to be adjusted both up and downwards
- An often used strategy is **'pricing to win'**
- A project cost is agreed on the basis of an **outline proposal**
- **Negotiations between client and customer** are then used to establish the real project specification
- The buyer and seller must **agree on what is acceptable**
- Often the project **costs are fixed** but the **requirements may be changed** to meet the costs



Source: <https://pixabay.com/en/price-tag-award-warranty-trailers-374004/>  
by geralt | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

7

## Software Project Planning



### Software Pricing

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the organization the opportunity to make a greater profit later. The experience gained may also help it develop new products.
Cost estimate uncertainty	If an organization is unsure of its cost estimate, it may increase its price by a contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract is awarded, high prices can be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a smaller than normal profit or break even than to go out of business. Cash flow is more important than profit in difficult economic times.

Source: Sommerville, I. (2011). Software Engineering, 9<sup>th</sup> Ed. Boston, MA, USA: Pearson Education Inc.

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

8

## Software Project Planning



### Plan-driven Development

- In a **plan-driven or plan-based** approach the development **process is planned in detail**
- This includes: **the work to be done, who will do it, the development schedule, and the work products**
- **'traditional'** way of managing large software development projects
- In contrast to **'agile'** development

**Pro:**

- Early planning allows for organizational issues to be closely taken into account

**Con:**


- Many decisions have to be revised due to changes in the environment



Source: <https://pixabay.com/en/office-startup-business-home-office-594132/>  
by StartupStockPhoto/CCO Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
9

## Software Project Planning




### Project Plans

Software project plans vary between organizations but usually include the following sections:

- **Introduction:** overall objectives
- **Project organization:** team and roles
- **Risk analysis:** risks and likelihood for them to occur
- **Hardware and software resource requirements**
- **Work breakdown:** activities, milestones, deliverables
- **Project schedule:** dependencies between activities etc.
- **Monitoring and reporting mechanisms**

**Note:** You may have a number of supplementary plans to support other process activities such as testing and configuration management.



Source: <https://pixabay.com/en/questions-who-what-how-why-where-2112217/>  
by geralt/CCO Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
10



## Software Project Planning



### The planning process

- Project planning is an **iterative process**
- The **initial plan** is drawn up during the startup phase
- As **more information becomes available**, you should regularly **revise the plan** to reflect requirements, schedule and risk changes
- **Changing business goals** lead to **changes in project plans**
- You need to **identify potential project constraints** (e.g. date, staff available, budget, tools, etc.)
- In conjunction with this you **identify project milestones and deliverables**
- The process then enters a **loop**
- You **review progress** and **note discrepancies**



Source: <https://pixabay.com/en/agenda-gantt-gantt-chart-project-153555/>  
by OpenClipart-Vectors | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

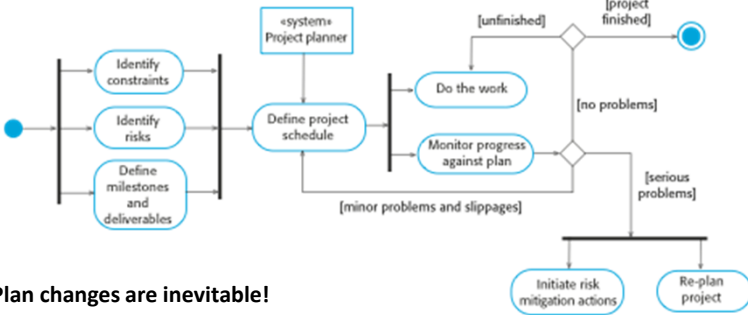
[office@mci.edu](mailto:office@mci.edu)

11

## Software Project Planning



### The planning process



**Plan changes are inevitable!**

Source: Sommerville, I. (2011). Software Engineering, 9th Ed. Boston, MA, USA: Pearson Education Inc.

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)


12

## Software Project Planning



### The planning process

- **Problems nearly always arise!**
- Your initial assumption and scheduling should therefore be **pessimistic rather than optimistic**
- Build in **sufficient contingency** so that there is **no need for re-negotiations** every time you go around a loop
- **Serious problems require risk mitigation** and may **require re-negotiations**
- If these actions fail you should call for a **technical project review**
- The outcome of the **review may be to cancel the project**
- Large software projects often **last for several years** where **objectives and priorities inevitably change**



Source: <https://pixabay.com/en/notes-paper-ball-memo-office-514998/>  
by congerdesign | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

13


## Software Project Planning



### Project Scheduling

Project scheduling is to **decide**

- How the work will be **organized in separate tasks**
- Estimate the **time** needed
- Estimate the **effort** required
- Define **who** will work on the tasks
- Estimate the **additional resources** needed (disk space, etc.)
- Estimate the **time** required on **special hardware** (e.g. simulator)
- Estimate the **travel budget**



Source: <https://pixabay.com/en/airport-travel-flight-times-690556/>  
by Free-Photos | CC0 Creative Commons

**Note:** The schedule is **rough** at the beginning (start-up phase) but **refined** throughout the project. It is necessary for both **plan-based and agile projects**.

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

14

## Software Project Planning




### Project Scheduling

**Agile project management**

- Here you use an **overall schedule** that identifies major phases of the project
- **Iterative scheduling** is then used for each phase

**Plan-driven project management**

- Here you break down the **total work** into **separate tasks**
- One task normally lasts **between 1 week and 2 month**
- Some tasks may be carried out **in parallel** where **different people** work on **different components**



Source: <https://pixabay.com/en/calendar-wall-calendar-days-date-1990453/> by webandi | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

15

## Software Project Planning




### Project Scheduling

You need to take into account the possibility that **things go wrong**

- People may fall ill
- Hardware may fail
- Parts may be delivered late

Calculate an **optimal schedule** and add an additional **contingency factor**

- Depending on the project you may add 30%-50%



Source: <https://pixabay.com/en/hurry-stress-time-management-2119712/> by TeroVesalainen | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

16



## Software Project Planning

### Schedule representation

- Project schedules may simply be represented in a **table** or **spreadsheet**
- Yet, this makes it **difficult** to see **relationships** and **dependencies** between different activities



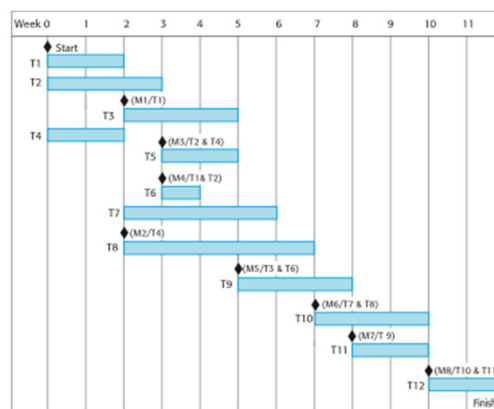
Source: <https://pixabay.com/en/airport-board-flying-scoreboard-1890943/>  
by andreas160578 [CC0 Creative Commons]

Alternative representations

- **Bar charts** (Gantt charts named after their inventor Henry Gantt)
- **Activity networks** (show dependencies between activities)


## Software Project Planning

### Schedule representation – Gantt Chart




Source: Sommerville, I. (2011). Software Engineering, 9th Ed. Boston, MA, USA: Pearson Education Inc.

## Software Project Planning



### Schedule milestones and deliverables

- You should also define **milestones** where **progress assessment** can be made
- Milestones should be **documented** by a **short report** (summarized progress made and work done)
- Deliverables are **special milestones** that produce **documents/products** to be **delivered to the customer**



Source: <https://pixabay.com/en/camino-santiago-path-milestone-174136/>  
by Donations\_are\_appreciated | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

19

## Software Project Planning



### Schedule task allocation

- After tasks are defined, **resources** need to be **allocated** to them
- The **key resources** in SE projects are the **software engineers** who will do the work
- **People** may be **working on more than one** project
- On the other hand they may be on **holiday**, attend **training courses**, or be **engaged in other activities**
- People working on several projects may **cause delays** in both of them as these **projects are linked** through these people



Source: <https://pixabay.com/en/application-business-collaboration-3428397/>  
by rawpixel | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)


20

## Software Project Planning



### Agile planning

- Agile software development follows an **iterative approach**
- Software is **developed and delivered to customers in increments**
- **Functionality is not planned in advance but decided during development**
- Increments depend on progress and on **customer priorities**
- Main argument: **priorities and requirements change** hence, we need more flexibility



Source: <https://pixabay.com/en/isolated-devops-business-3168537/>  
by diwou | CC0 Creative Commons

### 2 planning approaches

- Release planning: several month
- Iteration planning: 2-4 weeks for the team

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

21

## Software Project Planning



### Agile planning

- In Extreme Programming (XP) we call this **the planning game**
- It involves the **whole development team** including customer representatives
- **Specification is based on user stories**
- **Functionality** will be **missing** at the beginning
- Stories are **read and ranked**
- Too **large stories** are **broken into smaller ones**
- Stories are assigned **effort points**  
(e.g. 8 = difficult to implement; 2 = easy to implement)
- **Velocity** is the number of effort points implemented by the team per day



Source: <https://pixabay.com/en/web-web-developer-3935737/>  
by lakeyde | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

22

Software Project Planning


## Agile planning

- **Release planning** involves **selecting** and **refining** the **user stories** that have to be implemented **in one release**
- A release **date** is **chosen** and it is checked whether the **effort estimated allows for an on-time delivery**. If not, user stories are taken out or re-ordered
- **Iteration planning** is the first stage into the development process

**Note:** an iteration is complete, even if not all of the stories are implemented!



Source: <https://pixabay.com/en/concept-man-papers-person-plan-1868728/>  
by Perels | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
23

Software Project Planning


## Agile planning - challenges

- A major **difficulty** of **agile planning** is that it is reliant on **customer involvement** and availability
- Customers sometimes need to give **priorities to other work**
- It works **well with small teams**
- With **large teams** and especially **geographically distributed teams**, **traditional approaches** to project management are **more suitable**

**Note:** an iteration is complete, even if not all of the stories are implemented!



Source: <https://pixabay.com/en/strategy-chess-board-game-win-1080532/>  
by jarmoluk | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
24

## Software Project Planning



### Estimation techniques

- Project schedule estimation is **difficult!**
- It is usually based on **high-level user requirements definitions**
- Software may be novel where **aspects of it may be unfamiliar (e.g. hard and software)**
- The **skills of the team members** may be unknown
- Many **uncertainties** make it impossible to accurately estimate a schedule



Source: <https://pixabay.com/en/algebra-analyse-architect-1238860/>  
by morearmy | CC0 Creative Commons

**Two approaches:**

- Experience-based techniques
- Algorithmic cost modeling

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)


25

## Software Project Planning



### Experienced-based technique

- Rely on the manager's **experience of past projects**
- It usually helps to get a **group of people involved**
- Yet, new software **projects may not have much in common with previous projects**
- Software development **changes quickly** and so a **team may face unfamiliar techniques**



Source: <https://pixabay.com/en/library-books-knowledge-information-1147815/>  
by ElasticComputeFarm | CC0 Creative Commons

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15


[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

26



## Software Project Planning

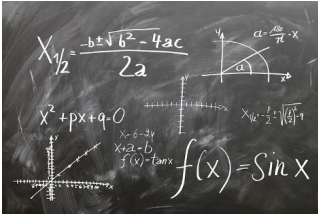


### Algorithmic cost-modeling

- Mathematical **formula** based on project size, software being developed, team, process and product features

**Effort = A x Size<sup>B</sup> x M**

**A...** constant factor depending on the organization  
**Size...** code size, functions, etc.  
**B...** Usually lies between 1 and 1.5 and addresses size and complexity (not linear!)  
**M...** Multiplier that combines process, product and development attributes



Source: <https://pixabay.com/en/mathematics-formula-physics-school-1509559/>  
by geralt (CCO Creative Commons)

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria

Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

27

## Software Project Planning



### Algorithmic cost-modeling - challenges

- It is often difficult to estimate **Size** in an early stage
- Also **B** and **M** are very **subjective**
- Estimations may also depend on **design decisions** that may not have been made
- Size estimations may also depend on the **programming language** e.g. Java may produce more lines of code than C
- Also, sometimes it may be possible to **reuse a significant amount of code** from previous projects
- You should use a range of estimations from **worst over expected to best**



Source: <https://pixabay.com/en/mathematics-calculator-formula-1453831/>  
by geralt (CCO Creative Commons)

DIE UNTERNEHMERISCHE HOCHSCHULE\*

MCI MANAGEMENT CENTER INNSBRUCK


6020 Innsbruck / Austria


Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)

[office@mci.edu](mailto:office@mci.edu)

28

Software Project Planning





## COCOMO II Model


- Empirical model derived by collecting data from a large number of software projects (Boehm, 1981). It supports the **spiral model** of development and embeds **sub-models**.

The sub-models of COCOMO II are:

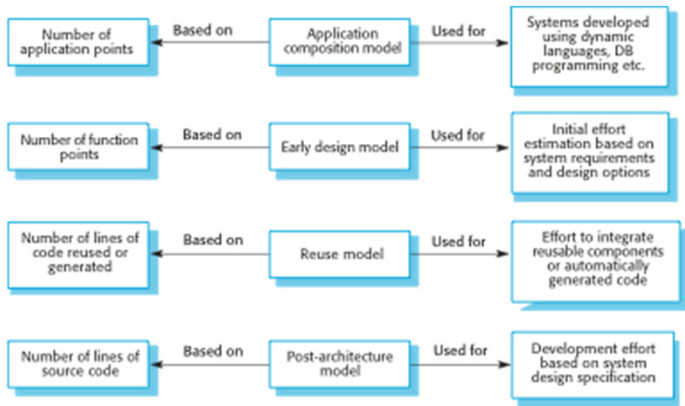
- **Application-composition model**  
Models the effort for creating systems from reusable components; estimates are based on **application points** (e.g. the number of screens, the number of reports, the number of modules, etc.)
- **An early design model**  
Used in the early stages of system design; estimates are based on **function points** (e.g. external inputs/outputs, user interactions, etc.) and translated into lines of code
- **A reuse model**  
Effort to integrate reusable components
- **A post-architecture model**  
After the architecture is designed a more accurate estimation can be made.

DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
29

Software Project Planning




## COCOMO II Model




```

graph LR
    subgraph "Application composition model"
        A1[Number of application points] -- Based on --> A2[Application composition model]
        A2 -- Used for --> A3[Systems developed using dynamic languages, DB programming etc.]
    end
    subgraph "Early design model"
        E1[Number of function points] -- Based on --> E2[Early design model]
        E2 -- Used for --> E3[Initial effort estimation based on system requirements and design options]
    end
    subgraph "Reuse model"
        R1[Number of lines of code reused or generated] -- Based on --> R2[Reuse model]
        R2 -- Used for --> R3[Effort to integrate reusable components or automatically generated code]
    end
    subgraph "Post-architecture model"
        P1[Number of lines of source code] -- Based on --> P2[Post-architecture model]
        P2 -- Used for --> P3[Development effort based on system design specification]
    end
  
```

Source: Sommerville, I. (2011). Software Engineering, 9th Ed. Boston, MA, USA: Pearson Education Inc.


DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK
6020 Innsbruck / Austria  
Universitätsstraße 15
www.mci.edu  
office@mci.edu
30

## Software Project Planning



### Project and staffing

- There is a **complex relationship between** the number of **people working on a project** and the **effort** that will be devoted to the project (i.e. person month)
- **Adding people** to the project **does not necessarily speed up** the progress accordingly
- **The more people** work on a project **the more time they spend on communicating and defining interfaces**
- In **large projects** this can lead to an **increase rather than a reduction** in the development schedule



Source: <https://pixabay.com/en/business-people-silhouette-teamwork-2089532/>  
by TeroVesalainen | CC0 Creative Commons


DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)  
[office@mci.edu](mailto:office@mci.edu)

31

## Summary



### What you should have taken away from this class:

- The **price** charged for a system **does not just depend on its estimated development costs** and the profit required by the development company. **Organizational factors** may mean that the **price is increased** to compensate for increased risk or **decreased** to gain competitive advantage.
- Software is **often priced to gain a contract** and the functionality of the system is then adjusted to meet the estimated price.
- **Plan-driven development** is **organized around a complete project plan** that defines the project activities, the planned effort, the activity schedule, and who is responsible for each activity.
- A **project milestone** is a predictable **outcome** of an activity or set of activities. At each milestone, a **formal report or progress should be presented** to management. A **deliverable** is a **work product** that is delivered to the project **customer**.
- **The XP planning game** involves the whole team in project planning. The plan is developed **incrementally** and, if problems arise, it is adjusted so that software **functionality is reduced instead of delaying the delivery** of an increment.
- **Estimation techniques** for software may be **experience-based**, where managers judge the effort required, or **algorithmic**, where the effort required is computed from other estimated project parameters.

DIE UNTERNEHMERISCHE HOCHSCHULE®  
MCI MANAGEMENT CENTER INNSBRUCK


6020 Innsbruck / Austria  
Universitätsstraße 15

[www.mci.edu](http://www.mci.edu)  
[office@mci.edu](mailto:office@mci.edu)

32

## References

- Beck, K. (2000). *Extreme Programming Explained*. Reading, MA, USA: Addison-Wesley.
- Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ, USA: Prentice Hall.
- Boehm, M. W., & Royce, W. (1989). Ada COCOMO and the Ada Process Model. *Proceedings of the 5th COCOMO Users' Group Meeting*. Pittsburgh, PA, USA.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Seattle, WA, USA: Microsoft Press.





DIE UNTERNEHMERISCHE HOCHSCHULE\*  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu

33





## Thank you for your attention!

mentoring the motivated.

DIE UNTERNEHMERISCHE HOCHSCHULE\*  
MCI MANAGEMENT CENTER INNSBRUCK

6020 Innsbruck / Austria  
Universitätsstraße 15

www.mci.edu  
office@mci.edu

34