



# Project Monitoring and Control

Software Engineering - Project Management

# What is PMC?



PMC is a group of processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project.

-Project Management Body of Knowledge (PMBOK)

# Purpose and implementation of PMC



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Purpose:

- To be proactive in finding issues ahead of time and taking corrective action.

# PMC Implementation



## Implementation:

PMC take place in parallel with Project Execution Process Group activities so that, while the project work is being executed, the project is being monitored and controlled by implementing the appropriate level of oversight and



# Agenda



- 
0. Introduction of CMMI and Scrum
  1. Monitor the Project Against the Plan
  2. Manage Corrective Actions to Closure
  3. Real Life Examples
  4. References
  5. Team

# 0. Introduction of CMMI and Scrum



## CMMI: Capability Maturity Model Integration

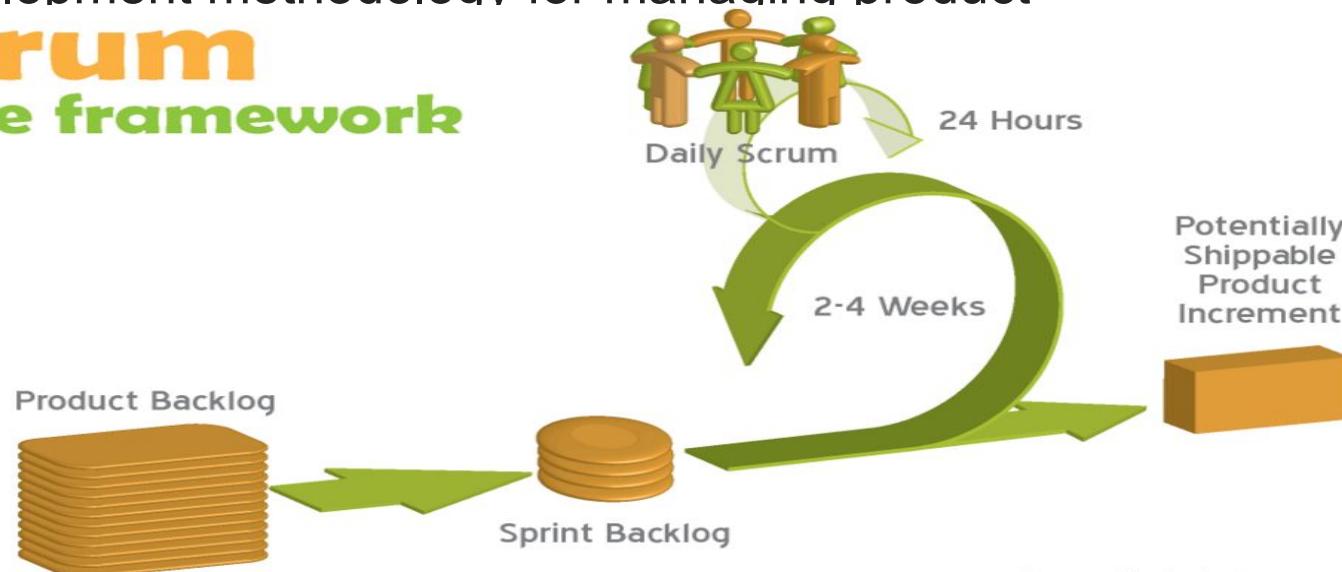
- A model that guides process improvement for system and Software engineering
- Used to evaluate the capabilities of an organization
- Guideline for process improvement

# 0. Introduction of CMMI and Scrum



**Scrum:** An iterative and incremental agile software development methodology for managing product

**Scrum**  
**the framework**



# 1. Monitoring the Project Against the Plan



- 1.1 Monitor project planning parameters.
- 1.2 Monitor commitments.
- 1.3 Monitor Project Risks.
- 1.4 Monitor Data Management.
- 1.5 Monitor Stakeholder Involvement.
- 1.6 Conduct Progress Reviews.
- 1.7 Conduct Milestone Reviews.



# 1.1 Monitor project planning parameters

## Parameters:

- Indicators of project progress and performance
- Attributes of work products and tasks
- Costs
- Effort and
- Schedule
- Etc.



# 1.1 Monitor project planning parameters

## CMMI:

Monitoring actual values of project planning parameters, comparing actual values to estimates in the plan, and identifying significant deviations.

## Scrum:

- Scrum board to check remaining efforts.
- Burndown charts to compare with planned and completed tasks.
- Charts to indicate progress of user stories/tasks.
- Charts to show status of individual tasks.



## 1.2 Monitor Commitments

- Comments and reviews on the ongoing or completed tasks.
- Usually done by scrum master and/or product owner.
- Predefined commitments for the tasks to be implemented.
- And reviews based on progress of the assigned tasks.



## 1.2 Monitor Commitments

### With CMMI:

- Commitments are mentioned in the project plan for each individual user stories.
- Team has to implement their tasks with the help of these comments.



## 1.2 Monitor Commitments

### With Scrum:

- Grooming and planning tasks before the beginning of the sprint.
- Daily standups, scrum meeting - status of the team.
- Burndown charts to view progress, pending tasks, their status and any impediments.
- Retrospective at the end of each sprint.



## 1.3 Monitor Project Risks

### Required Process of Implements

- Monitoring Residual risk
- Identifying new risk
- Risk response plan
- Tracking identified risk
- Mainly evaluating the risk effectiveness of the process throughout the project.

# 1.3 Monitor Project Risks



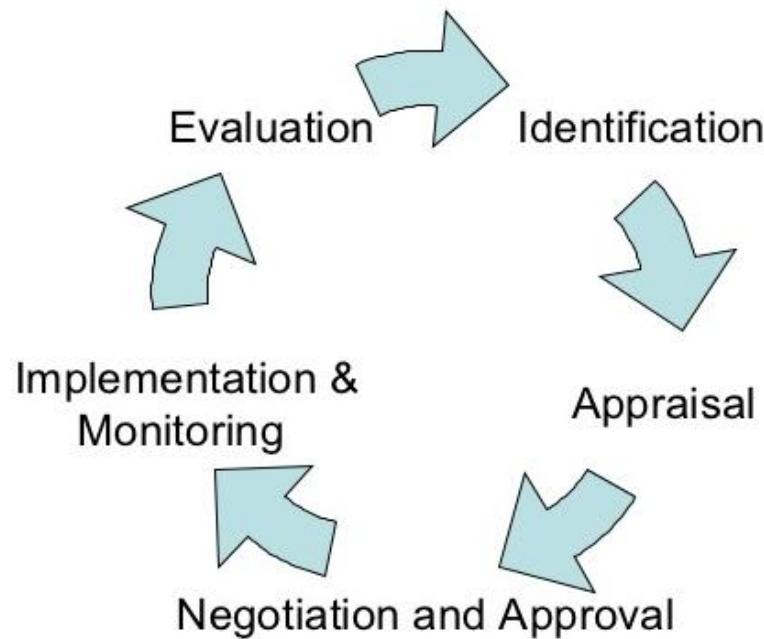


## 1.3 Monitor Project Risks

### With CMMI:

- Monitoring the project against the plan.
  1. Frequent monitoring of project parameters.
  2. Tracking project risk and commitment.
  3. Ensuring stakeholders involvement.
- When varian begin, immediately corrective measures will be taken to get the project back on track.

# 1.3 Monitor Project Risks



# 1.3 Monitor Project Risks



## With Scrum:

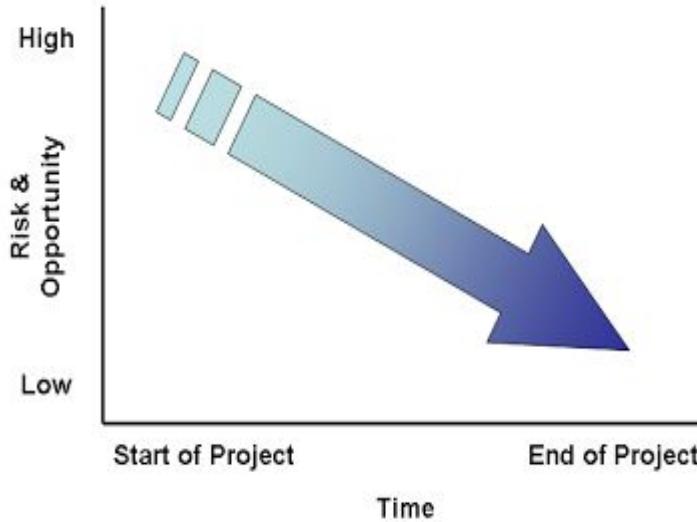
- Risks are identified on daily basis.
- Severity of the risk is assessed based on the impact of the undesired outcome.
- Risk can still not be analysed completely in this stage.

# 1.3 Monitor Project Risks

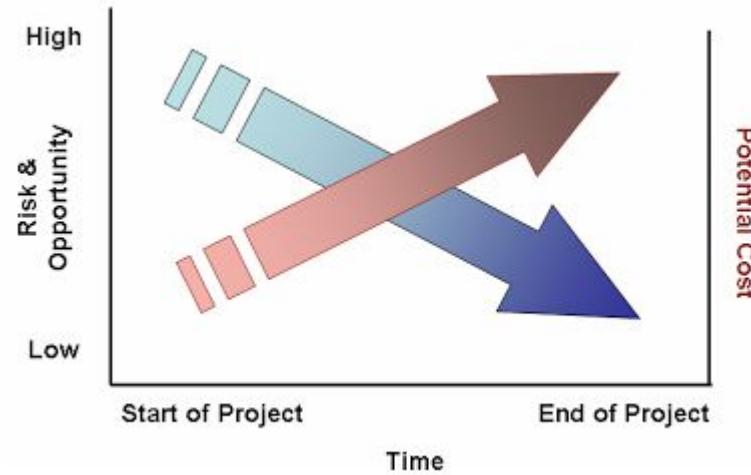


# 1.3 Monitor Project Risks- Example

Expected



Actual Result

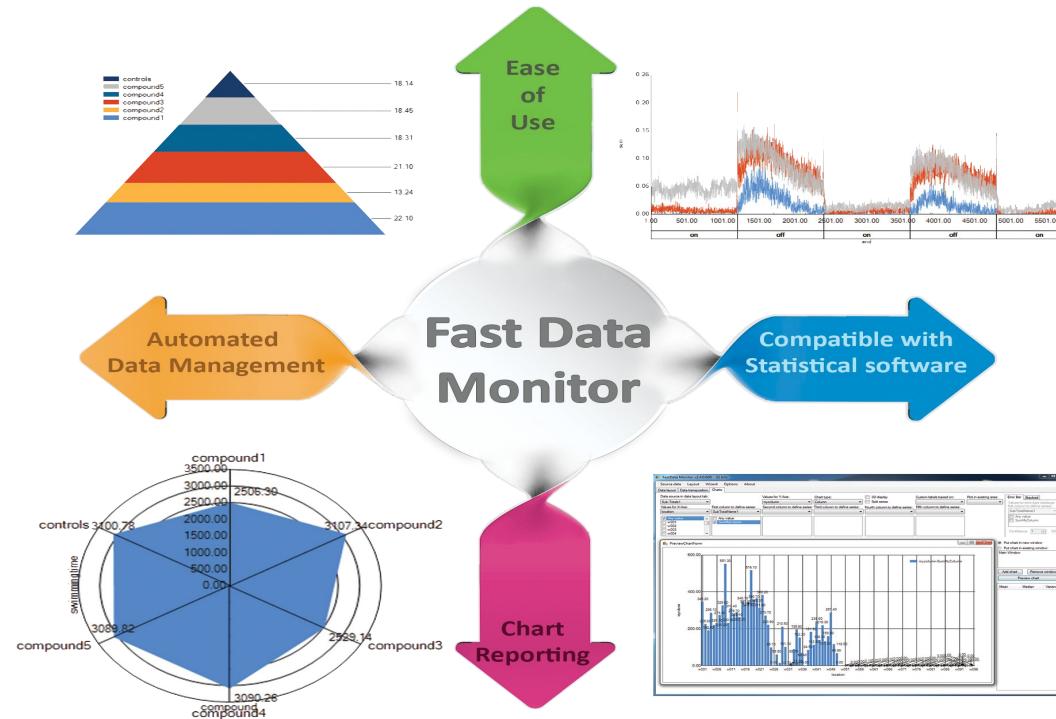




## 1.4 Monitor Data Management

- Collection, storage, processing, dissemination and efficient use of information in the context of monitoring and evaluation.
- It is the routine process of tracking inputs and outputs.
- It should provide information on whether an intervention is on budget and on track.
- Evaluating systematic and objective assessment at one point.

# 1.4 Monitor Data Management





## 1.4 Monitor Data Management

### With CMMI:

- Stakeholder involvement should be monitored.
- Conducting Process review.
- Eg: Records of Supplier Involvement.

# 1.4 Monitor Data Management



# 1.4 Monitor Data Management



## With Scrum:

- Daily Scrum Meeting.
- Sprint Review meeting.



# 1.4 Monitor Data Management



# 1.5 Monitor Stakeholder Involvement



- Stakeholder involvement should be monitored periodically to ensure that appropriate interactions occur.
- It may be necessary to re-plan stakeholder involvement depending on the results of monitoring and changes in project requirements.





# 1.5 Monitor Stakeholder Involvement

## CMMI:

- Monitor stakeholder involvement against the project plan

## Scrum:

- Scrum master involves Stakeholders when required

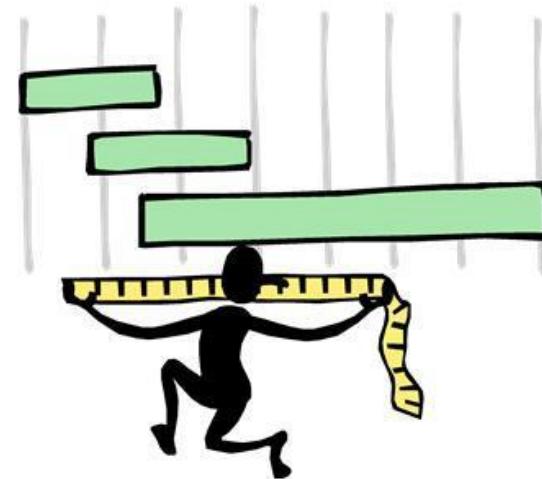
# 1.6 Conduct Progress Reviews

## CMMI:

- Periodically review the project's progress, performance, and issues

## Scrum:

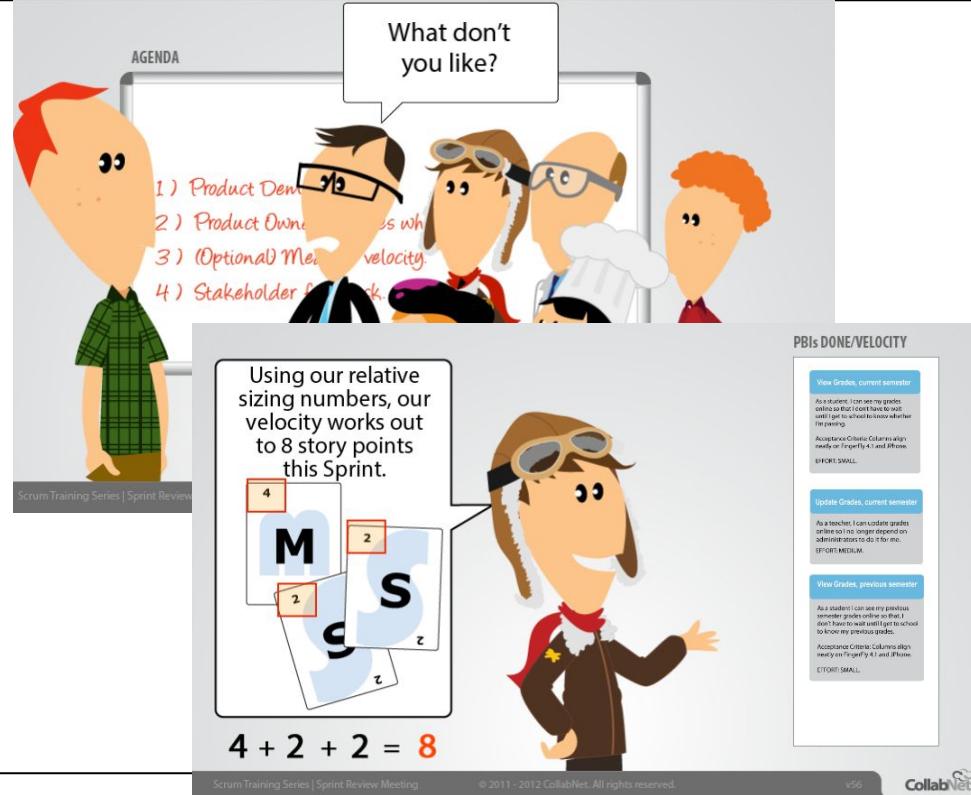
- Discussions on team commitments at
  - Daily Scrum meeting
  - Sprint review meeting
- Retrospectives



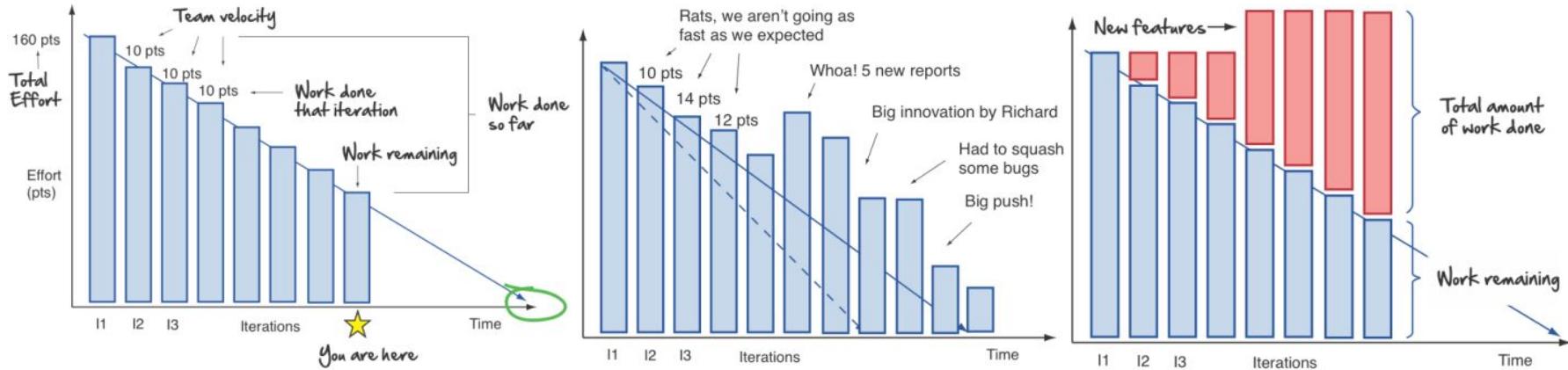
# 1.7 Conduct Milestone Reviews

## CMMI:

- Accomplishments & results at pre-planned points.
- Track action items.

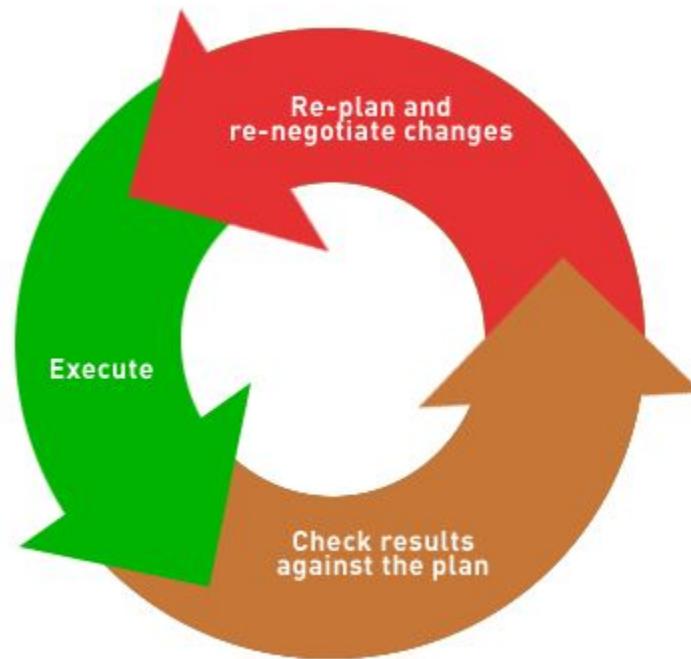


# Excursion: Burndown-Chart



- Ideal.
- Realistic.
- Personal expansions.

## 2. Manage Corrective Actions to Closure



## 2. Manage Corrective Actions to Closure



**How to get aware of derivations?**

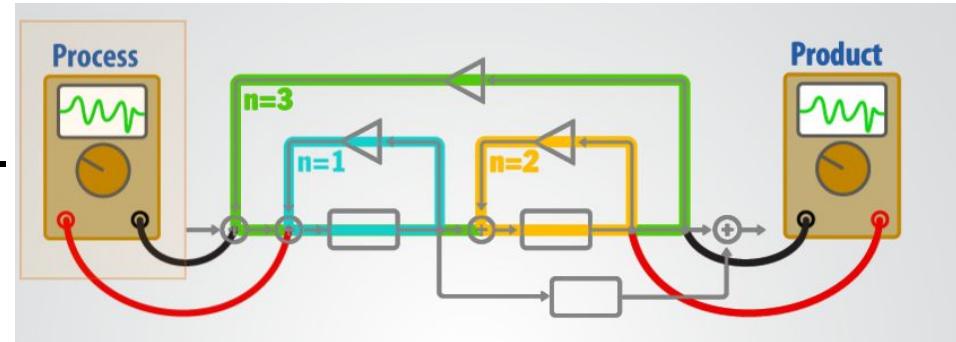
**Analyze Issues!**

## 2.1 Analyze Issues



### CMMI:

- Identify significantly derivation.
- Determine Corrective Actions.



### Scrum:

- Daily Scrum.
- Sprint Retrospective.
- Impediment Backlog.



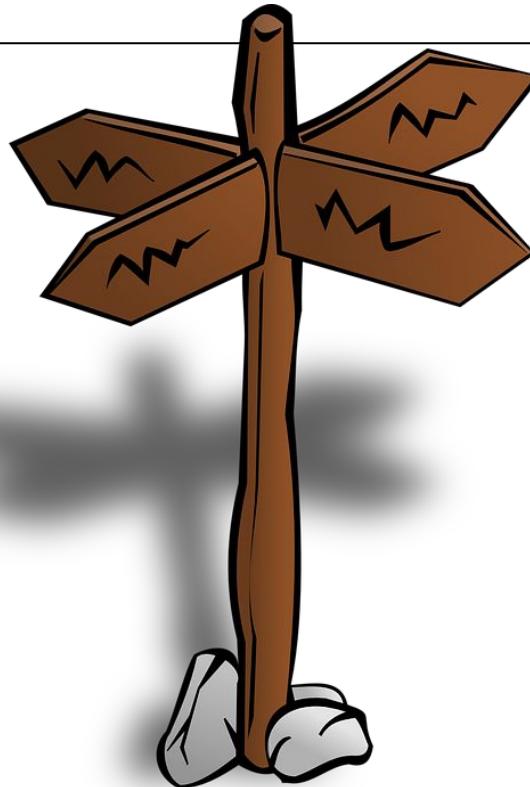
## 2.2 Take Corrective Action



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

### How to take corrective action ?

## 2.2 Take Corrective Action



add

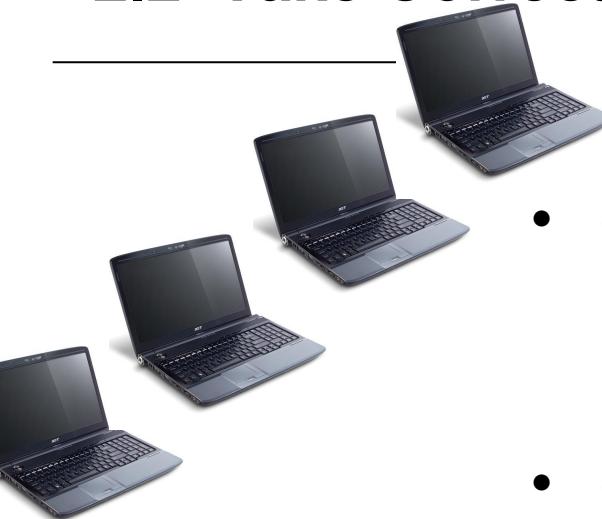


adjust & recalibrate



reduce

## 2.2 Take Corrective Action - add



- **add resources**
  - hardware
  - software/licences
  - environment (e.g. office, air condition)
- **add team member**
  - split up tasks
  - reduce stress
- **add knowledge**
  - training
  - new team member



## 2.2 Take Corrective Action - reduce



- **remove resources**
  - unused licenses
- **reduce tasks**
  - reduce goals
  - (reduce documentations)
- **remove team member**
  - bad work results
  - bad influence on team motivation
- **Remove distracting things and situations!**
  - e.g: dogs in the office



## 2.2 Take Corrective Action - adjust



- **check resources**
  - limited resources used by others (e.g licenses)
- **check team structure**
  - behavior
  - change team compositions
- **check tasks**
  - spread tasks again
  - reestimate tasks
  - change approach
- **check environment**
  - change office
  - working hours



## 2.2 Take Corrective Action



### CMMI

1. document the appropriate actions
2. review solution
3. inform stakeholder and get agreement
4. negotiate changes to all



### Scrum

1. daily scrum & sprint review
  - discuss solution with team
  - apply solution in team



## 2.3 Manage Corrective Action



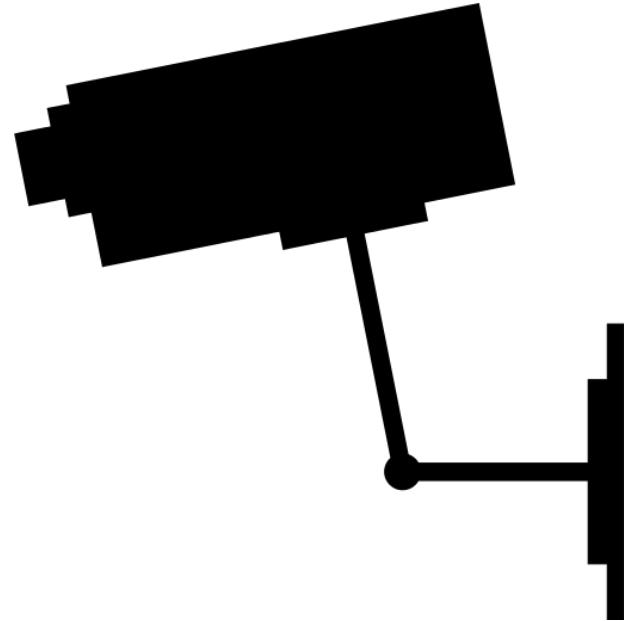
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

### How to manage corrective action?

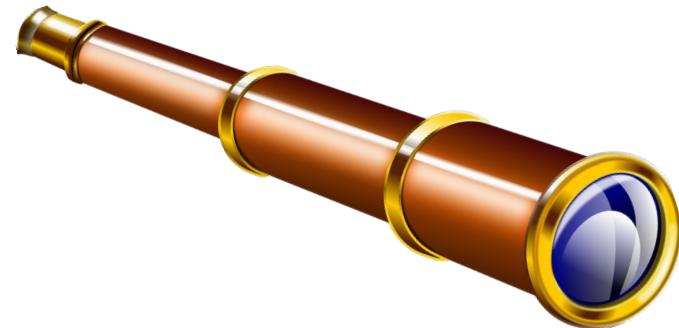
## 2.3 Manage Corrective Action



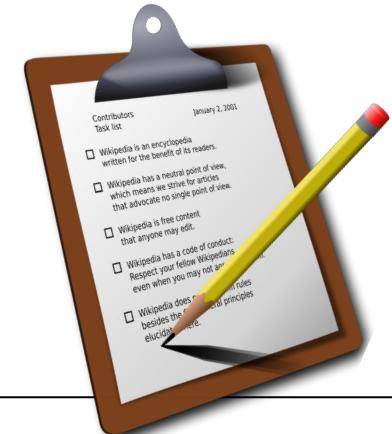
Monitor progress!



## 2.3 Manage Corrective Action



- analyse results
- check against plan
- check against schedule
- define new alert points e.g:
  - amount of overtime
  - tasks per weeks
  - average errors in code
  - cups of coffee



## 2.3 Manage Corrective Action



cups of coffee ?



## 2.3 Manage Corrective Action



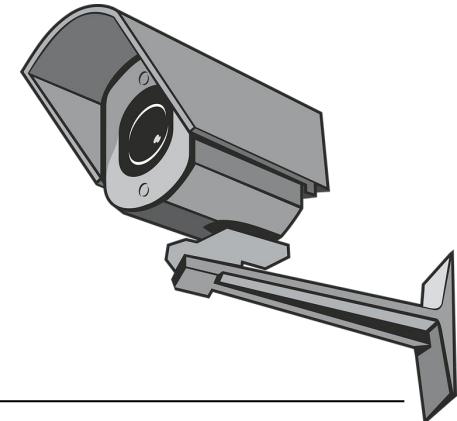
### Scrum



- daily scrum
  - daily status
  - correct results
  - positive feedback
- sprint review
  - reached scrum points

### CMMI

- correct results
- checking against plan



## 2.3 Real Life Examples

---



**some examples ...**

## 2.3 Real Life Examples



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



### 3. Real Life Examples - Solutions



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

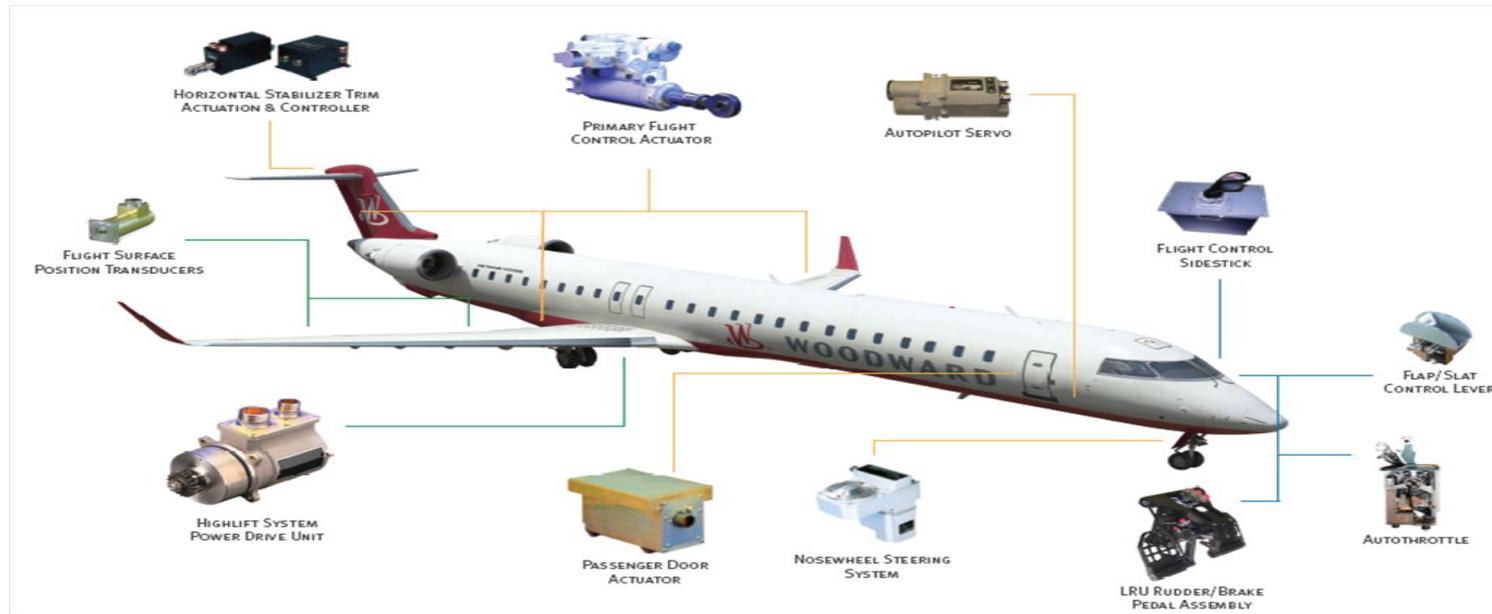


### 3. Real Life Examples - Solutions



### 3. Real Life Example of PMC(2)

Good knowledge about the system makes things easier.



# PMC Parameters



- Know about the factors which affects the system.
- whereabouts of risk and their time of occurrence.
- Planning to avoid identified risks and controlling them leads to successful results.

# Without Monitoring and Control



- Unexpected End Results
- Failure of the system

- Uncertainty.
- Loss of confidence.
- Decrease of Quality.



### 3. Real Life Examples(3)

---

- Super market needs a constant check on there frequently moving products.



### 3. Real Life Examples(3)

Without monitoring and control:

- Drop in sales (Loss)
- Loss of Customers.
- Loss of Stakeholders.



## 3.2 Real Life Example: Alright



ÜBERSICHT

TICKETS

STATISTIKEN

KUNDEN

ADMIN

Angemeldet als [User]

OTRS

Übersicht

Erinnerungs-Tickets

Meine gesperrten Tickets (0) | Tickets in meinen Queues (0) | Alle Tickets (0)

keine

Eskalierte Tickets

Meine gesperrten Tickets (0) | Tickets in meinen Queues (0) | Alle Tickets (0)

keine

Neue Tickets

Meine gesperrten Tickets (0) | Tickets in meinen Queues (0) | Alle Tickets (30)

keine

Offene Tickets / Beantwortung erforderlich

Meine gesperrten Tickets (0) | Tickets in meinen Queues (0) | Alle Tickets (4)

Zum Anfang der Seite

The screenshot shows the OTRS ticket management system interface. At the top, there's a navigation bar with links for ÜBERSICHT, TICKETS, STATISTIKEN, KUNDEN, ADMIN, and a search bar. The main area is divided into several sections: 'Übersicht' (Overview) with 'Erinnerungs-Tickets', 'Eskalierte Tickets', 'Neue Tickets', and 'Offene Tickets / Beantwortung erforderlich'; '7 Tage-Statistik' (7-day statistics) with a line graph showing ticket volume over the week; and 'Online' status for agents and customers. A footer at the bottom indicates the system is powered by OTRS 3.1.10.

## CANCOM

- Inhouse-Development with Scrum for ERP-System

# Question & Answer



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT





## 4. Team

- Saumweber, Florian (2354534)
- Wauer, Axel (2199990)
- Praveen Kumar, Pendyala (2919474)
- Shivakumar, Tejamurthy (2311784)
- Anusha Ravichandran (2521040)



## 5. References

1. CMMI® for Development, Version 1.3 CMMI-DEV, V1.3  
<https://www.sei.cmu.edu/reports/10tr033.pdf>
2. <https://de.wikipedia.org/wiki/Scrum>
3. <http://scrumtrainingseries.com/SprintReviewMeeting/SprintReviewMeeting.htm>
4. <http://scrumtrainingseries.com/SprintRetrospectiveMeeting/SprintRetrospectiveMeeting.htm>
5. <http://scrumtrainingseries.com/SprintRetrospectiveMeeting/SprintRetrospectiveMeeting.htm> Slide 4 and 46
6. <http://www.agilenutshell.com/burndown>
7. <http://scrumtrainingseries.com/SprintReviewMeeting/SprintReviewMeeting.htm> Slide 48 and 107
8. <http://www.scrumguides.org/scrum-guide.html>