

# The problem management process flow

Theodor Panagiotakopoulos, Hellenic Open University



Co-funded by the  
Erasmus+ Programme  
of the European Union



# Aim and objectives

- This presentation focuses on the problem management process describing its main phases and respective activities.
- The **objectives** of this presentation are to:
  - Present the main phases of managing problems
  - Describe the activities performed in every phase of the problem management process

- At the end of this presentation, you will be able to:
  - Distinguish problem management from incident management
  - Identify the types of problem management
  - Recognize the outcomes of problem management

# Table of contents

- Section 1 <Problem identification>
- Section 2 <Problem control>
- Section 3 <Error control>

# The three phases of problem management

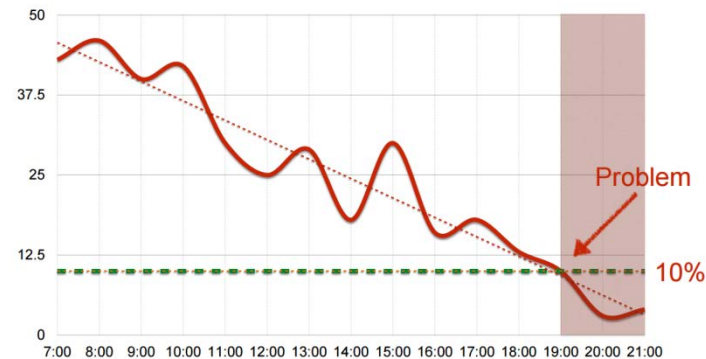
Problem  
Identification

Problem  
Control

Error Control

# Problem identification

- The **first step** in the problem management process is to identify that a problem exists, or, in other words, **detect** it
- Problems may be raised either **reactively** in reaction to incidents or **proactively**
- The **inputs** to reactive problem management can come from a number of sources (e.g. incident management, event management and the service desk)
- **Proactive** problem management will identify problems often before any incidents occur



Source: <https://www.zabbix.com>

# Ways to detect problems

- Performing **trend analysis** of incident records
- Detecting **duplicate** and **recurring** issues
- During major incident management, identifying a **risk** that an incident could recur
- Analyzing information received from **suppliers** and **partners**
- Analyzing information received from **internal** software developers, test teams, and project teams



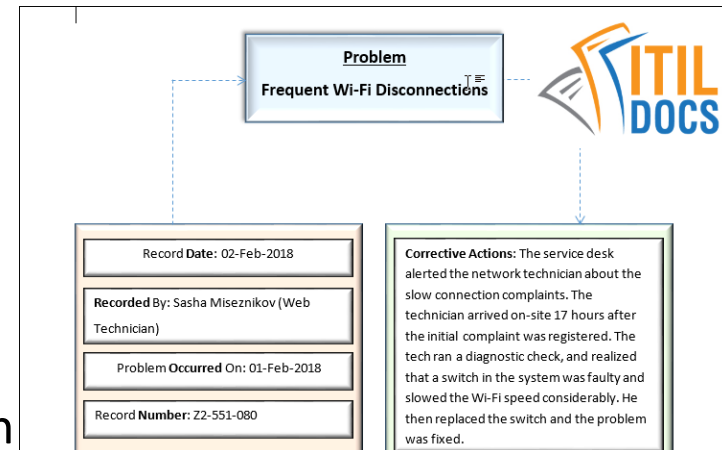


- Having identified that a problem exists, a **problem record** should be logged
- The problem record must contain **all** the **relevant information**, time-stamped to provide a complete picture
- It is crucial that the **full details** of the problem are recorded
- This will allow **analysis** to take place and will enable comparisons to be made between problems



Source: <https://logentries.com>

- All **incidents** caused by the problem should be **linked to** the **problem record**
- Some tool sets enable the creation of a problem record from an incident, with automatic linking between the two
- **Typical details** entered in a problem record include:
  - ✓ incident record number
  - ✓ date and time
  - ✓ priority and category
  - ✓ details of who reported it and when
  - ✓ details of the service and equipment used
  - ✓ a description of the incident and actions taken



Source: <https://www.ital-docs.com>



# Problem control



Co-funded by the  
Erasmus+ Programme  
of the European Union

- Problem **control activities** include **problem analysis**, and **documenting workarounds and known errors**
- Just like incidents, problems will be **prioritized** based on the risk they pose in terms of probability and impact to services
- When analyzing incidents, it is important to remember that they may have **interrelated causes**, which may have complex relationships
- When a problem cannot be resolved quickly, it is often useful to find and document a **workaround** for future incidents



Source: <https://www.ital-docs.com>

# Categorizing problems

- Problems should be **categorized** in the same way as incidents
- it is recommended that the **same system** is used **as** adopted by the **incident management** process for any particular organization
- Correct and meaningful categorization will allow helpful **metrics** to be produced
- An essential **prerequisite** for identifying trends in incidents is the accurate and consistent categorization of incidents
- If every service desk analyst logs the same fault **differently**, it will be impossible to discern a trend

## Prioritizing problems

- As with incidents, the **priority** of a problem should be based on
  - the **impact** to the business of the incidents that it is causing and
  - the **urgency** with which it needs to be resolved
- The problem manager should also consider the **rate of reoccurrence** of the incidents that occur

It is possible that a “frozen screen” that can be resolved with a reboot is not a high-priority incident

if it is occurring 100 times a day, the combined impact to the business may be severe, so the problem needs to be allocated a high priority

# A simple problem prioritization system

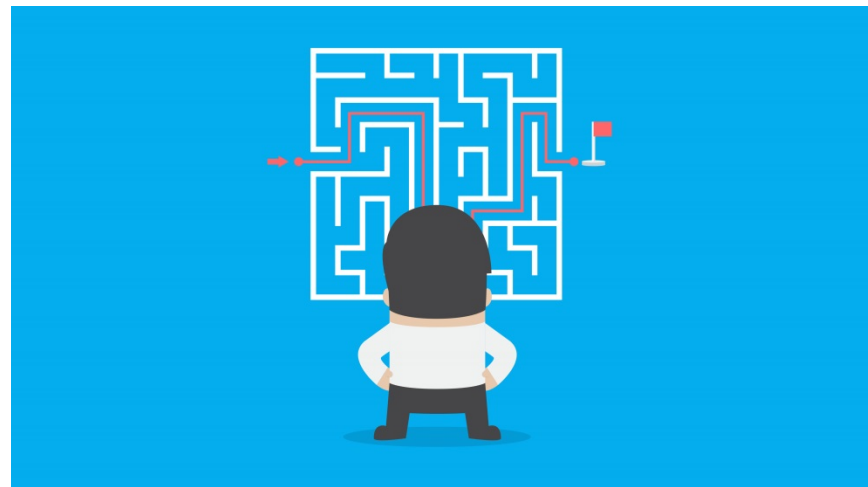
		Impact		
		High	Medium	Low
Urgency	High	A	B	C
	Medium	B	C	D
	Low	C	D	E

Priority	Description	Resolution time
A	Critical	1-2 hours
B	High	8-10 hours
C	Normal	Up to 24 hours
D	Low	1-2 days
E	Very low	Schedule

- The next stage in the process is to **investigate** and **diagnose** the problem
- There may not be the resources to investigate every problem, so the **priority level** assigned to each will govern which ones get the necessary attention
- It is important to **allocate resources** to problem investigation, because until the problem is resolved, the incident will recur, and resources will be spent on incident resolution
- The **aim** of the investigation and diagnosis phase is to ascertain the root cause of a problem



- The specific problem analysis method used is less important than that it is **thorough** and **systematic**
- The more common **techniques**, which will be described in detail in the next presentation, include:
  - Chronological
  - Flowcharting
  - Brainstorming
  - Affinity diagram
  - Fault isolation
  - Rapid problem resolution



Source: <https://www.projectmanager.com>

## Identifying a workaround

- Although the aim of problem management is to find and remove the **underlying cause** of incidents, this may take some time
- **Meanwhile**, the incident or incidents continue, and the service is affected
- When a user suffers an incident, the **first priority** is to restore the service so that they can continue working
- A priority of the process, therefore, is to provide a **workaround** to be used until the problem is resolved



- The workaround does not fix the underlying problem, but it **allows the user** to continue working by providing an alternative means of achieving the same result
- The **workaround** can be **provided to** the **service desk** to enable them to resolve the incidents, while work on a permanent solution continues
- The details of the **workaround** are **documented within** the **problem record**, which remains open, since the fault still exists and is continuing to cause incidents

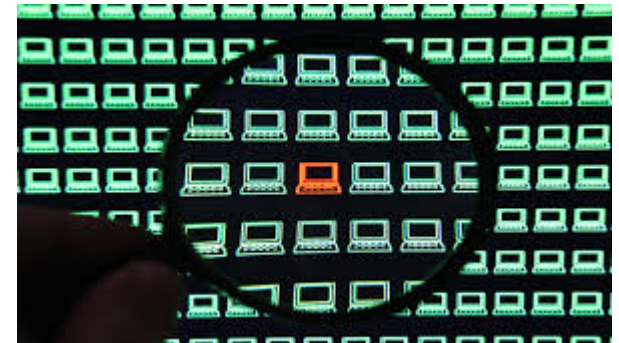


Source: <http://nahbnow.com>

- An **effective incident workaround** can become a permanent way of dealing with some problems, where permanent resolution of the problem is not viable or cost-effective
- In this case, the problem remains in the **known error** status, and the documented workaround is applied when related incidents occur
- Indicative **reasons** why workarounds remain in place for some time:
  - a permanent fix is too **risky**
  - a permanent fix is too **costly**
  - the business **impact** of the problem is not significant enough to justify further diagnosis at this time
  - the problem will be permanently fixed by a **new release** that is currently being planned

## Raising a known error record

- A **known error record** should be raised when the diagnosis has been completed and especially when a workaround has been identified.
- Information about all known errors, including which problem record it relates to, is kept in **the known error database** (KEDB)
- The known error database is an important source of information for the **service desk** and **support groups** handling incidents and problems
- When repeat incidents occur, the **support staff** can refer to the KEDB for the workaround



Source: <https://www.bmc.com>



# Error control



Co-funded by the  
Erasmus+ Programme  
of the European Union

- Error control **activities** manage known errors, with the goal of converting known errors into solutions and removing them from the KEDB
- Where a permanent solution requires **change control**, this has to be analyzed from the perspective of cost, risk and benefits
- Error control also regularly **re-assesses the status of known errors** that have not been resolved, taking account of the overall impact on customers and/or service availability, the cost of permanent resolutions and effectiveness of workarounds
- The **effectiveness of workarounds** should be evaluated each time a workaround is used

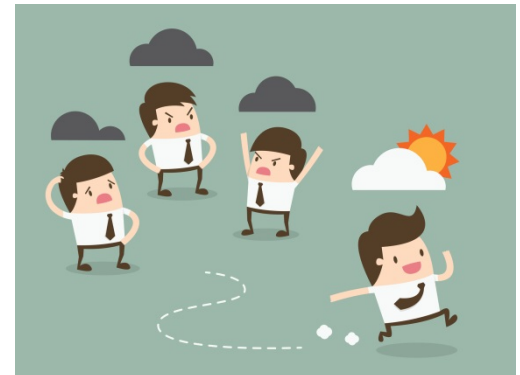
- When problem management has **identified a solution** to the problem, it should be implemented to resolve the underlying fault
- Implementing the resolution may involve a degree of **risk**, however
- So, the **change management** process will ensure the risk and impact assessment of the requested change is satisfactory before allowing any change
- Ultimately, the decision whether to go ahead with the resolution despite the risk is a **business decision**
- The business damage of the problem may mean the business is prepared to **accept** the risk in order to have the fix implemented



- When a **permanent solution** has been identified, tested, and implemented, the problem record can be updated and closed
- Any open **incidents** caused by the problem can be **closed** too
- The **KEDB** should be **updated** to show that the problem is resolved, so any future incidents will not have been caused by it
- However, the information contained within the problem record may prove useful in **addressing** a future, **similar problem**



- Each organization should define what constitutes a **major problem**
- This may be all priority one problems, anything above a particular priority level that continues or some other **criteria**
- Once a major problem has been resolved, a **review** should be held to identify any lessons that can be learned from what occurred
- The **importance of good recordkeeping** is apparent at a review; a well-documented problem, with a full history of steps taken, will provide useful information



Source: <https://www.fool.com>

## The goal: lessons learned

- It is crucial that these reviews look at **lessons learnt** rather than becoming 'allocation of blame' sessions
  
- Specifically the major problems review should **examine**:
  - ✓ Those things that were done correctly
  - ✓ Those things that were done wrong
  - ✓ What could be done better in the future
  - ✓ How to prevent recurrence
  - ✓ Whether there has been any third-party responsibility and whether follow-up actions are needed.

- ITIL (2011). ITIL Service Operation. London: TSO
- Brewster E., Griffiths R., Lawes, A., and Sansbury J. (2012), IT Service Management. A Guide for ITIL Foundation Exam Candidates, 2<sup>nd</sup> ed. British Informatics Society
- Gallacher, L., & Morris, H. (2012). ITIL Foundation Exam Study Guide. John Wiley & Sons.
- Axelos (2019). ITIL® Foundation, ITIL 4 edition, TSO



Mobile and Pervasive Computing, Quality and Ambient Intelligence Laboratory

Hellenic Open University, Patras, Greece

Email: [panagiotakopoulos@eap.gr](mailto:panagiotakopoulos@eap.gr)

Theodor Panagiotakopoulos was born in Greece in 1981. He received his Diploma and PhD from the Department of Electrical and Computer Engineering, University of Patras, Greece in 2006 and 2011 respectively. His research interests include, among others, pervasive computing, internet of things, ambient intelligence, mobile health and ambient assisted living systems, telemedicine and biomedical applications. Until now, he has published over 25 articles in international conferences and journals, as well as in international book chapters. He has participated in 7 National and European R&D projects focusing on IoT and e-Health, as well as on the development of educational content for digital skill acquisition in various application sectors via e-learning programs. Since 2016, he is an adjunct assistant Professor at the Department of Electrical and Computer Engineering of University of Patras.

## Credits

- Author: Theodor Panagiotakopoulos, HOU
- Technical Reviewers: Christos Pierrakeas and Panagiota Polymeropoulou, HOU
- Scientific Reviewers: Theodoros Grassos and Eleni Damianou, AKMI S.A.



[www.project-musa.eu](http://www.project-musa.eu)



[musa@daissy.eap.gr](mailto:musa@daissy.eap.gr)



@MuseumSectorAlliance



#MuseumSectorAlliance



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0)

**Project Number: 575907-EEP-1-2016-1-EL-EPPKA2-SSA**



Co-funded by the Erasmus+ Programme of the European Union



This project has been funded with support from the European Commission. This presentation reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

