ENGR 301 Lecture Notes

Scope and Time management are PMBOK knowledge areas which are crucial for project success. GitLab, while having many strengths for managing a project, has features which tend to mix scope and time and makes it more difficult than perhaps it should be to manage each — particularly as it's critical to have an excellent (not perfect, just excellent) definition of project scope *before* the work to meet scope is planned in time.

PMBOK Scope and Time

It's worth reminding ourselves of what PMBOK advises we should be doing within the scope and time knowledge areas during the planning phase.

Scope Management

Defines, and gains agreement and common understanding with stakeholders, on the work required to complete the project successfully.

Tools:

- Scope statements
- Work breakdown structures
- Requirements analysis
- Statements of work
- Scope management plans

Work Breakdown Structures

A Work Breakdown Structure (WBS) is a useful tool for defining project scope, it is a document that breaks all the work required for the project into discrete tasks, and groups those tasks into a logical hierarchy. Often shown in two different forms:

- Graphical or chart form
- Tabular or list form

Work Packages: are deliverables at the lowest level of the WBS, where they can be appropriately assigned to and managed by a single accountable person. Each work package should be defined in enough detail to estimate what it would cost and how long it would take to create.

Time Management

Estimates how long it will take to complete the work, and develops an acceptable project schedule. Aims for timely completion of project!

Tools include:

- Activities
- Milestones
- Gantt charts

Activities

An activity is a distinct, scheduled portion of work performed during the course of a project.

The goal of the defining activities process is to ensure that project team members have a complete understanding of all the work they must do as part of the project scope so that they can start scheduling the work.

Rationale: how can you estimate how long it will take or what resources you need to prepare a report if you don't have more detailed information on the report?

An Activity List breaks Work Packages into Activities, with scheduling and relationship information, to manage project time. Typically benefits complicated projects the most.

Milestones

In PMBOK, a milestone is a significant event on a project.

- Milestones help identify necessary activities, and with sequencing, scheduling and monitoring.
- The greater the degree of interdependence between Work Packages, the greater in number (and more accurate) the milestones will be.
- Milestones, naturally, often occur between some (but not necessarily all) nodes in a WBS.

Gantt Charts

We all know about these:

- ubiquitous;
- Closely related to the WBS and will use the same terms;
- shows the project schedule as a series of related timelines;
- helps identify and resolve busy periods;
- helps with project monitoring and control.

See: Project Management Graphics (or Gantt Charts) https://bit.ly/4cer0Fc for a *very* full discussion of the

strengths and weaknesses of Gantt charts.

GitLab Features

GitLab provides *Milestone*, *Epic* and *Iteration* features which can be used to manage scope and time.

- *Iterations* are composed of *Issues* at the level of the GitLab Project
- *Milestones* are composed of *Issues* at the level of the GitLab Project
- *Epics* are composed of *Issues* at the level of the GitLab Group

The course methodology recommends *Milestones* be preferred over *Iterations*; *Epics* are best-used to obtain a *Roadmap* (Gantt Chart) but are not well-suited to composing or decomposing a WBS due to their flat presentation of Issues contained within the *Epic*.

GitLab provides several features of *Issues*, *Milestones* and *Epics* for managing and tracking time, but poor separation of scope from time:

- Issues have due dates, time estimates and time expenditure;
- Milestones have start and end dates;
- Iterations have a cadence;
- Epics have start and end dates;

GitLab's features use several different terminologies and concepts from several different sources. Managing scope and time in GitLab is not as straight-forward as it could be. Advice:

- Constructing work packages as Milestones is good practice;
- Constructing Milestones corresponding to iterations (little "i") is also good practice;
- Mirroring Epics from Milestones will help with a Roadmap view;
- Iterations are probably best avoided in favour of Milestones

Using these features is important: without time management via Milestones (and Epics), it will be hard to know whether or not project goals will be achieved.
