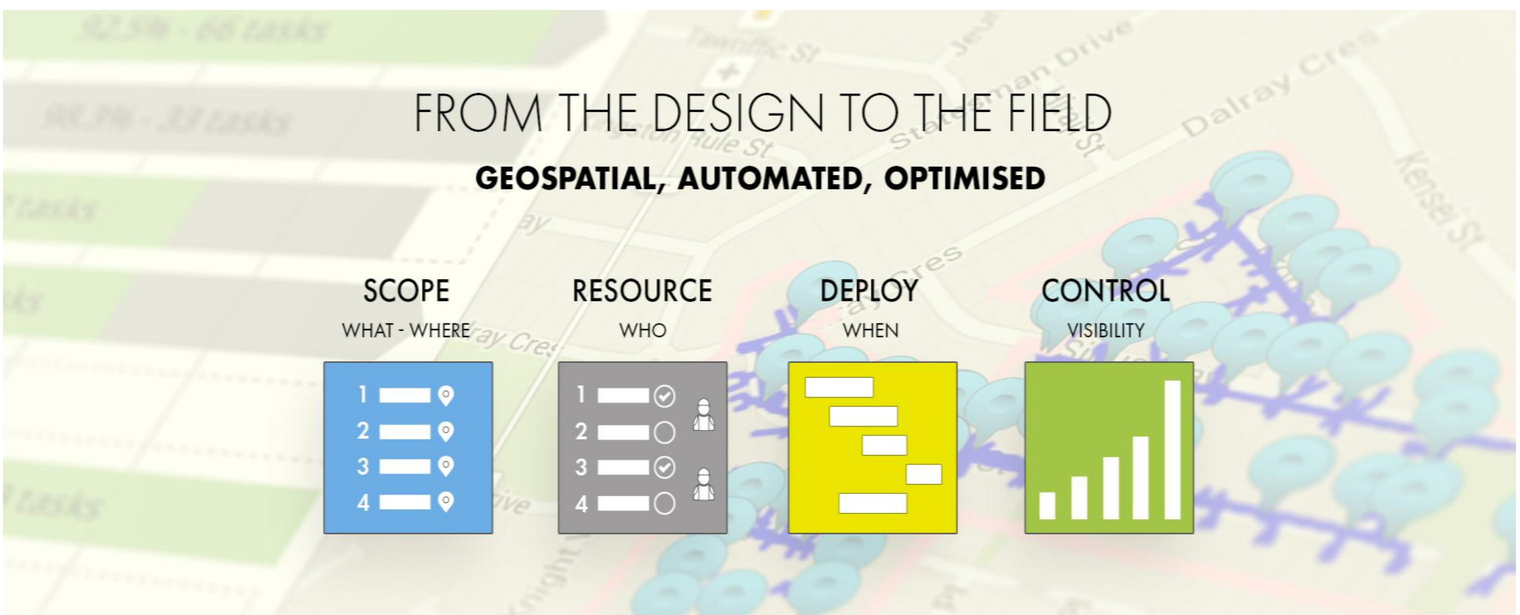
Achieving project management agility through automation

John.Podlena@rendernetworks.com.au

RENDER
DEPLOYMENT MANAGEMENT

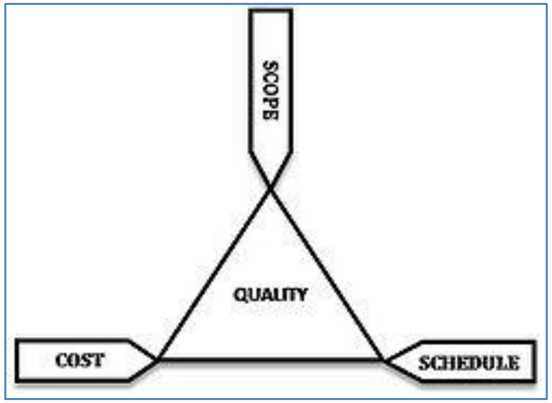


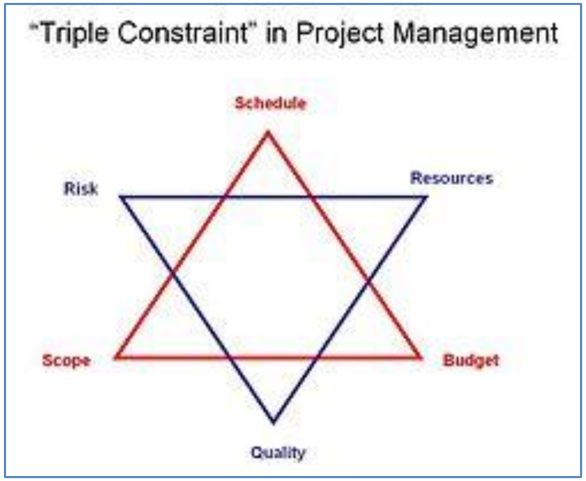
Agenda

- Project management concepts:
  - Critical paths, PERT/estimation
  - Resource levelling
  - EVM, s-curves
- Case study: deploying software to manage an NBN Fibre to the Premises rollout
  - Key PM & software challenges
  - Development and rollout approach
  - Outcomes

Project management constraints

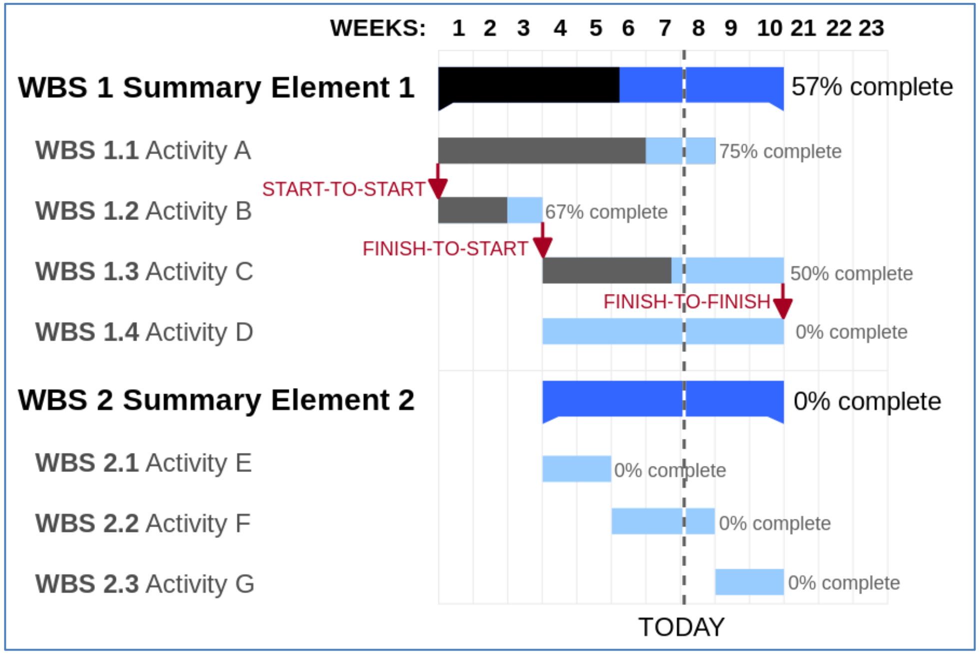
- Typically a project planner needs to balance constraints on scope (incl. quality), time and cost.
- Often one or more of these constraints are fixed prior to project planning.





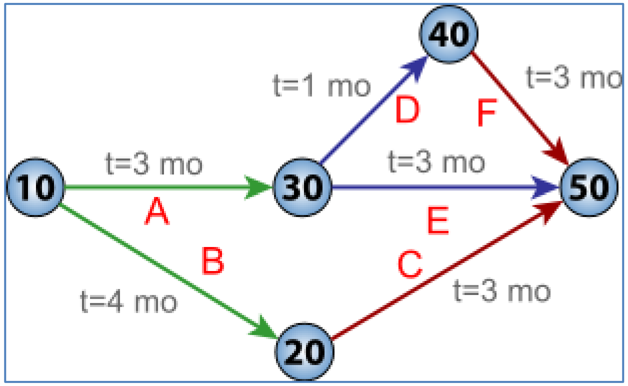
Project management constraints

- In creating a project schedule, a PM is attempting to understand how achievable the scope and time elements of the project constraints are.
- In assigning resources and materials to a schedule, the PM can estimate cost.
- Ideally, the sequencing and resourcing against project tasks can be optimised against time and/or cost.



Optimisation: the critical path method (CPM)

- Aims to identify those tasks that, if late, will make the whole project run late.
- Algorithm (for F-S tasks) is based on calculating the Earliest Start Time, EET, LST and LET, and then identifying the longest path (often tasks with zero "total float").
- Tasks on the critical path can be "crashed" – e.g. assigning more resources to bring them in earlier.
- Assumes resources are relatively unlimited (some extensions allow for resources).



Task	Duration	EST	EET	LST	LET	Total Float = LST-EST
A	*3*	0	3	0	3	0
B	*4*	0	4	0	4	0
C	*3*	4	7	4	7	0
D	*1*	3	4	3	4	0
E	*3*	3	6	4	7	1
F	*3*	4	7	4	7	0

Program evaluation and review technique (PERT)

- Aims to address uncertainty in project scheduling (e.g. provide better estimates).
- Typically uses the CPM.
- Using estimates for:

  a = best-case duration

  b = worst-case duration

  m= most likely duration

  then:

  Expected duration $\mathbf{E}(\mathbf{a}, \mathbf{b}, \mathbf{m}) = (\mathbf{a} + 4\mathbf{m} + \mathbf{b}) / 6$

  Variance $\mathbf{V}(\mathbf{a}, \mathbf{b}) = (\mathbf{b} - \mathbf{a})^2 / 36$

- Can then ask "what is the probability that the project will be completed in N days"?
- Relies on a bunch of assumptions:
  - Activity durations are independent, and follow a beta distribution.
  - The critical path found through CPM will always be the critical path.

Sources of estimation error

- Initial estimates are made against many random variables including:
  - Productivity
  - Environment
  - Other human factors:
    - Student syndrome: tasks are started at the latest possible start (possibly negating the use of schedule “buffers”).
    - Parkinson’s law: "work expands so as to fill the time available for its completion”.
    - Human multi-tasking: “perform multiple tasks at the same time equally poorly” (context switching, counter to deep concentration).
- Often estimation improves over the life of the project.

Resource levelling

- The process of assigning resources to tasks is known as **resource levelling**.
- We could define the resource levelling problem space as a set **R** of **x** indexed by:

  $\mathbf{x}_{\mathbf{tjr}}$ (time slot **t**, job **j**, resource **r**) where **x** = {0 | 1},

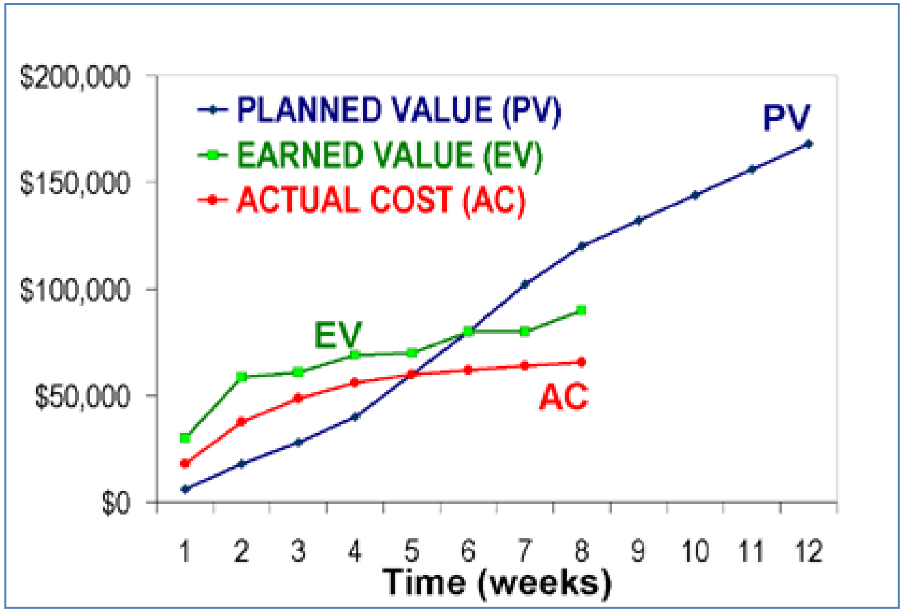
  subject to all manner of constraints (e.g. covering tasks taking more than one time-step needing to be contiguous).
- Can be very similar to a **Job Shop Scheduling** problem (with resources in the place of machines and travel time between jobs in the place of machine set-up times).

Resource levelling

- May want to optimise to:
  - Minimize makespan (schedule duration)
  - Minimize cost (which may competes with using more resources to crash critical path)
  - Minimize site/location changes
- Can solve this assignment problem using Integer Programming; larger problems may require some decomposition or grouping first.
- Can solve using "greedy" algorithms – for example focusing on resourcing tasks with less total float (e.g. those close to/on the critical path); these may be quicker but potentially can miss optimisation opportunities (esp. in resource-poor situations; doing longer tasks earlier can cause resource bottlenecks).

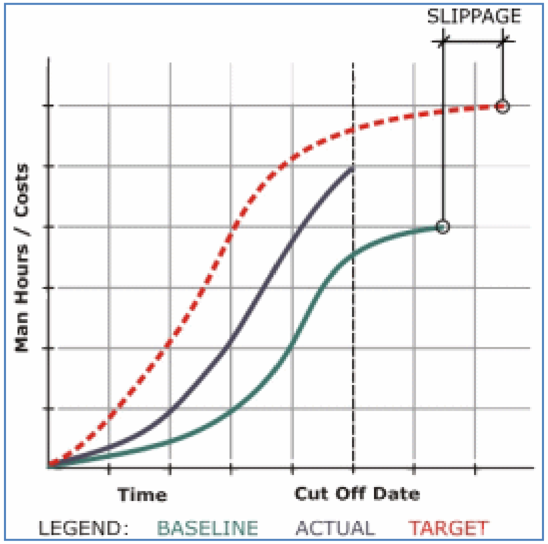
Monitoring progress: Earned Value Management

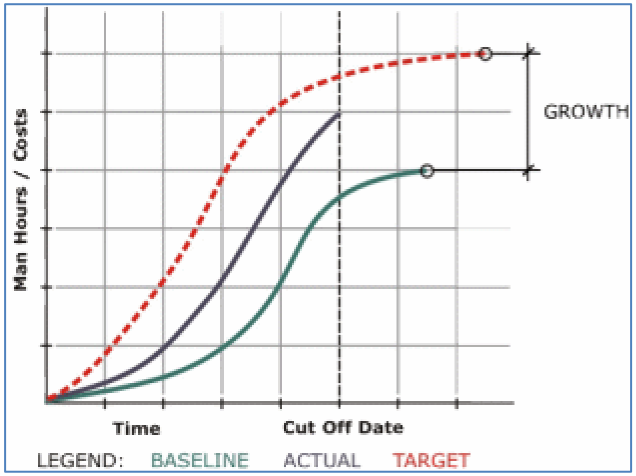
Allows progress to be managed over time using a planned value (as scheduled), an Earned Value (e.g. the value of assets actually delivered) and Actual Cost (often directly related to effort).



Monitoring progress: "S-curves"

- Shows a project schedule's cumulative baseline, target (re-scoped) and actual as costs (or hours) over time.
- Slippage and growth can be shown as the target schedule is updated.



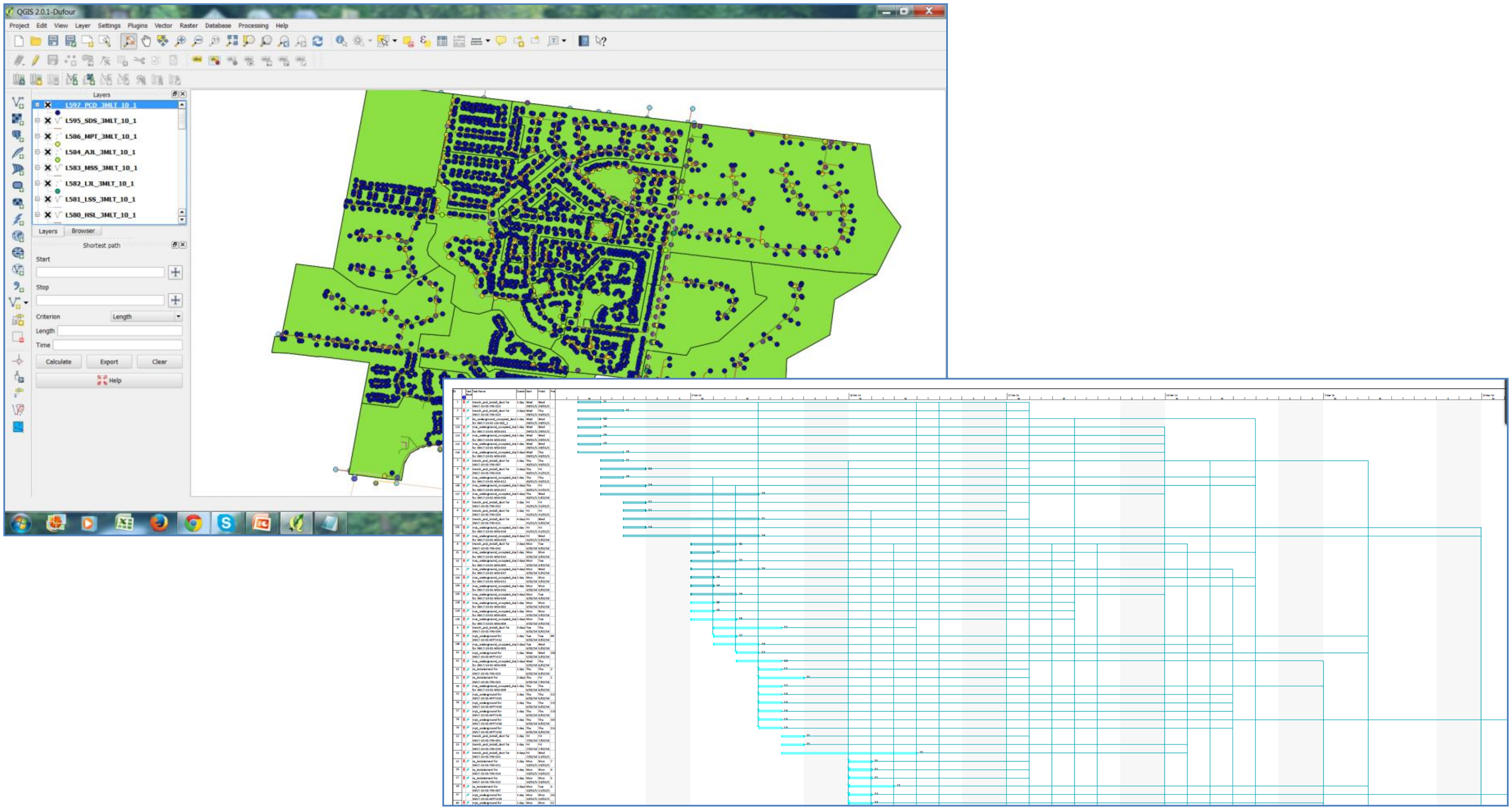


Case study: develop & rollout software to manage an NBN rollout

- Main aim of the new software (“Render”) was to address issues such as:
  - A lack of detail in project schedules (due to efforts to reduce PM workload) degrading estimation and true project status reporting.
  - Rescheduling being costly (not automated).
  - Critical path activities not being necessarily realised and/or prioritized.
  - The impact and therefore prioritization of delay and blocked tasks not being clear.
  - Poor task coordination and poor alternative task identification in the field (low utilisation) resulting from unclear task dependencies (sequencing).
- The software was to be trialled at a site in Melton, Victoria

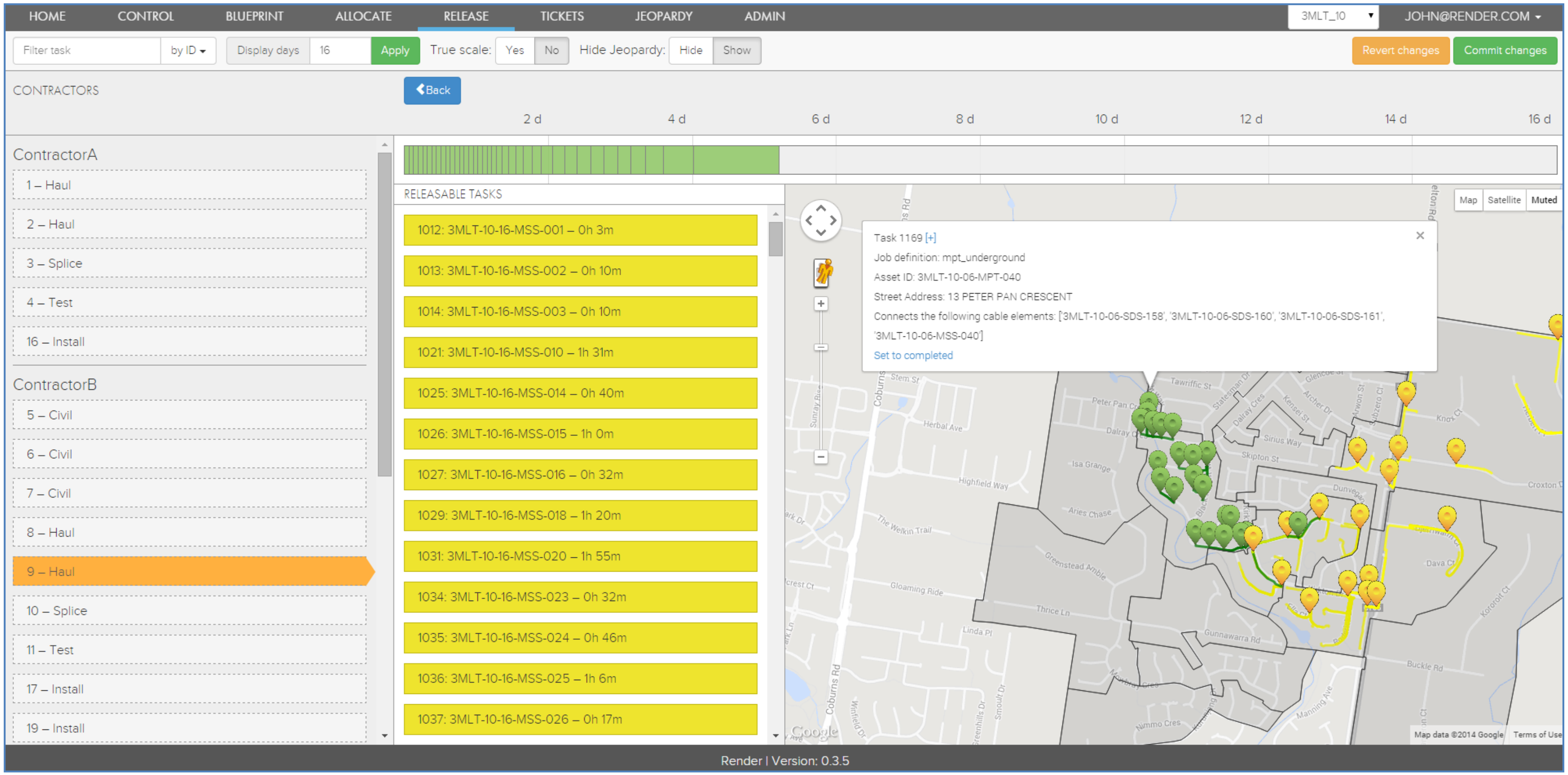
Key functionality

- Derive a full set of tasks from Geospatial design data, including a task dependency hierarchy (reduce PM schedule bootstrapping workload).



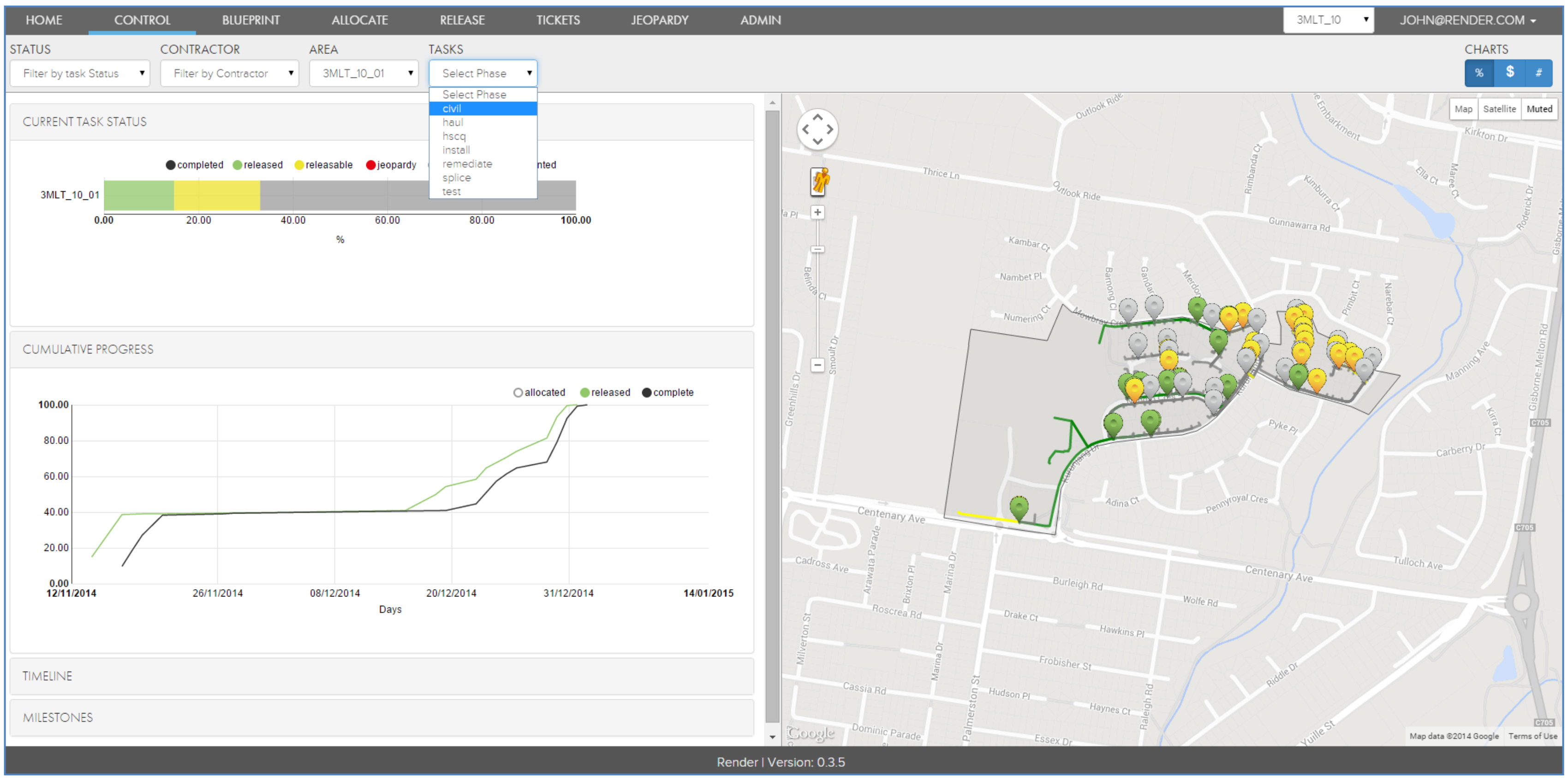
Key functionality

- Schedule work down to the resource level based on optimisation criteria and constraints (e.g. minimise makespan, group like tasks geospatially and temporally).
- From the schedule, give the resources a set of tasks for only the next week for them to “tactically” choose from (all with predecessor tasks complete) - allowing for working around blocked tasks.



Key functionality

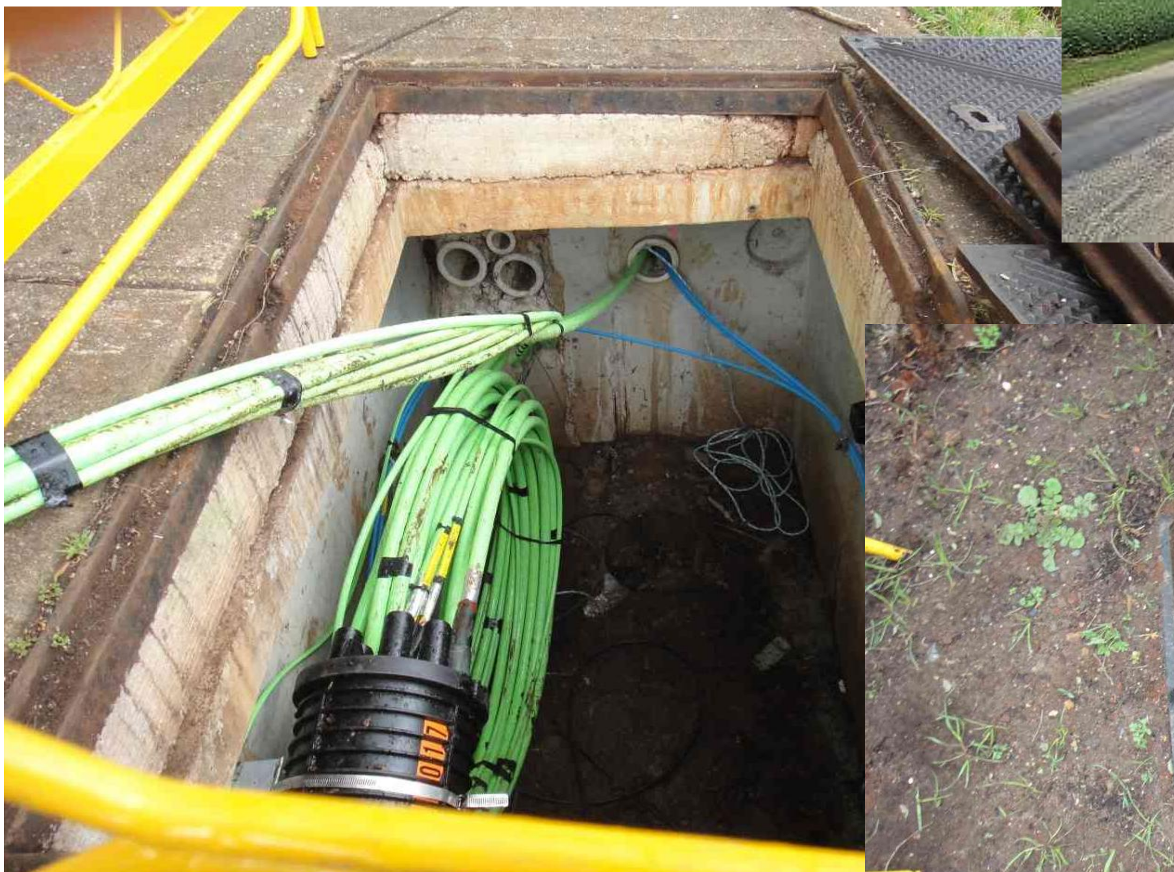
- Reschedule every night based on new data from the field.
- Update estimates as actual productivity information is returned to the system.
- Provide a well-designed reporting tool showing past, current and forecast project states down to the task level if required (including output ready for EVM & S-curve analysis).



Challenges: cardinality

- 4000+ tasks
- 50 work crews (3 subcontractors)
- 5 main phases:
  - civil (e.g. trench digging, pipe installation)
  - Haul (pull fibre cables of varying size through duct)
  - Splice (splice thousands of fibre joins)
  - Test (test fibre connections)
  - Install (connect fibre from the street to each premises)
- Technical risk was mitigated by:
  - Developing a “greedy” scheduling engine first for an early MVP
  - Following up with a divide-and-conquer approach to derive sub-problems solvable as IP formulations

Civil and hauling work



Splicing and testing

Challenges: acceptance

- Management needed to trust that Render was guiding the build in the correct direction, and reporting on the true progress, current and forecast state of the project.
- In-field management and resources needed to trust that the tasks assigned were appropriate.

Challenges: acceptance

In order to gain and retain acceptance, we:

- Ran at first in parallel with the existing project management process. *Render was used increasingly as the build continued.*
- Added a simulation engine to the software, simulating tasks being completed each day followed by the rescheduling and release of tasks. *This helped us maintain a high level of software quality.*
- Allowed for a “manual override” of the tasks generated by the scheduling engines. *As the results became more acceptable, management moved to use the fully automated approach.*

Outcomes

- The Melton trial of the Render software was considered a success, with a significant reduction in overall makespan (~61%) and cost (~50%) compared to similar rollouts.
- The software quickly gained acceptance from the users in its ability to allow work to continue around disruptions.
- The offline forecasting/scheduling functionality was used to model different resourcing options very quickly.

Outcomes

- In the few cases where the system was not used, teams often turned up to do jobs they couldn't start, and in one case multiple teams turned up to do the same job.
- ***The biggest value to the business was seen to be:***
  - ***Render's automated & optimised reactivity around disruption***
  - ***The system-enforced, strict adherence to maintain task dependencies and ownership in the field***