

**Timelines:  
When to do what  
with an SMP?**



*"The best time to plant a tree is  
20 years ago.*

*The second best time is now."*

# Summary



[www.software.ac.uk](http://www.software.ac.uk)

- Why does project timing affect how you view planning?
- When best to revise a Software Management Plan?
- How to start a Software Management Plan?

# The need to address change



[www.software.ac.uk](http://www.software.ac.uk)

- A lot of things can change over time
  - Stakeholders, research goals, development practices, use of infrastructure
- SMP: evolving document to reflect and inform action
  - Are we doing the right things for the right reasons for the right people? *What needs to change?*
  - Ideally, detailed practices emerge from this

# How does "When" change things?



[www.software.ac.uk](http://www.software.ac.uk)

- At any stage of a project, there is
  - What you're doing already
  - What you have, should, or want to do later
- Three major time points to consider
  - Before or when project starts, in the middle, near the end
  - SMP still an intention of what you will do
  - Perspective will be different

# In the beginning...



www.software.ac.uk

- Consider and *involve* your stakeholders early
  - Who, why, what - but also how?
  - SMPs clarify expectations
- Can't do everything
  - Prioritise
  - Start small, build over time

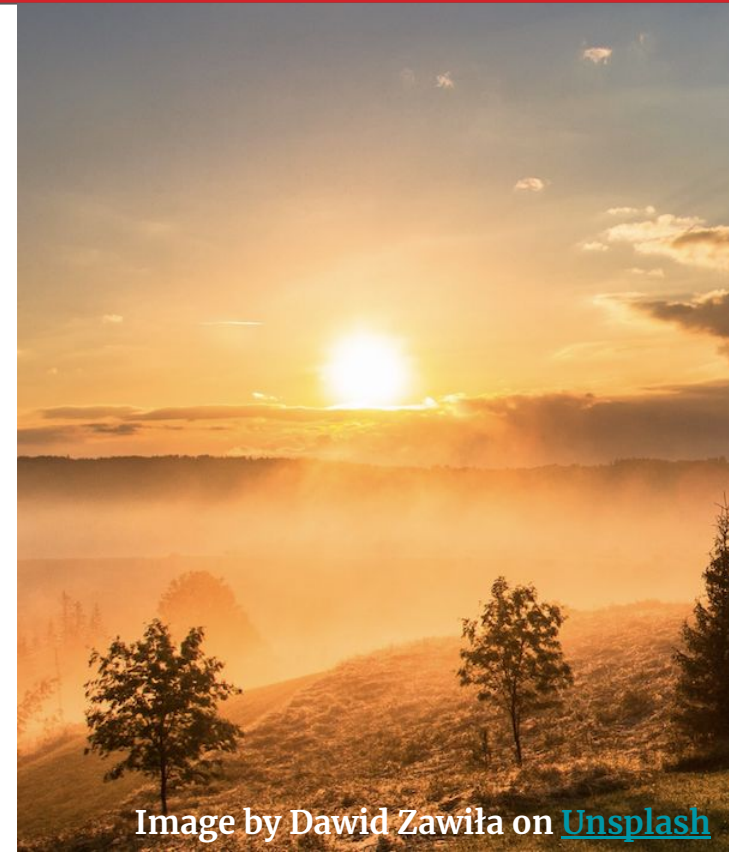


Image by Dawid Zawila on [Unsplash](#)



# End of the road...



www.software.ac.uk

- AKA "sunsetting"
- Often seen as negative
  - A "step" in a bigger story
  - Software has intrinsic value
- In project teams
  - *Product* changes, not *project*
- What to persist/preserve, in what form, and how?



Image by Call Me Fred on [Unsplash](https://unsplash.com/photos/CallMeFred)



# How does this fit with managing projects?



[www.software.ac.uk](http://www.software.ac.uk)

- As PI or project manager?
- *When and how often, where, and to what extent you do Software Management Planning?*

## Consider

- Nature of software and development model, e.g.
  - *One-off purpose* - no support, archived or preserved
  - *Traditional waterfall* - long release cycles
  - *Agile* - short, iterative cycles
- Team size - *small vs large*

# Example timeline - how focus changes



www.software.ac.uk

## 1. At the start - minimal

What's being produced/managed/shared?  
Who's doing what?

## 3. Sunsetting

Revisit produced/managed/shared  
What should persist/be preserved, and how?  
...and for who?

v1.0

v2.0

v3.0

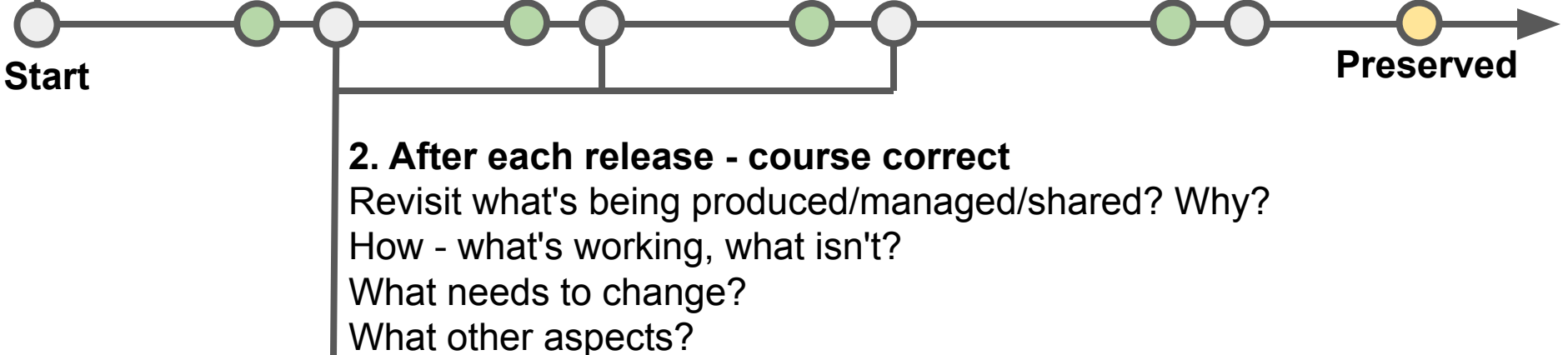
v4.0

End

Preserved

## 2. After each release - course correct

Revisit what's being produced/managed/shared? Why?  
How - what's working, what isn't?  
What needs to change?  
What other aspects?





*What are you doing now? What should you do next?*

## **Goal: to create...**

- Your first minimal draft SMP, based on current understanding on what you're doing now
- A rough timeline to introduce desired changes later (e.g. the next release, or other suitable deadline)



*What are you doing now? What should you do next?*

## Questions...

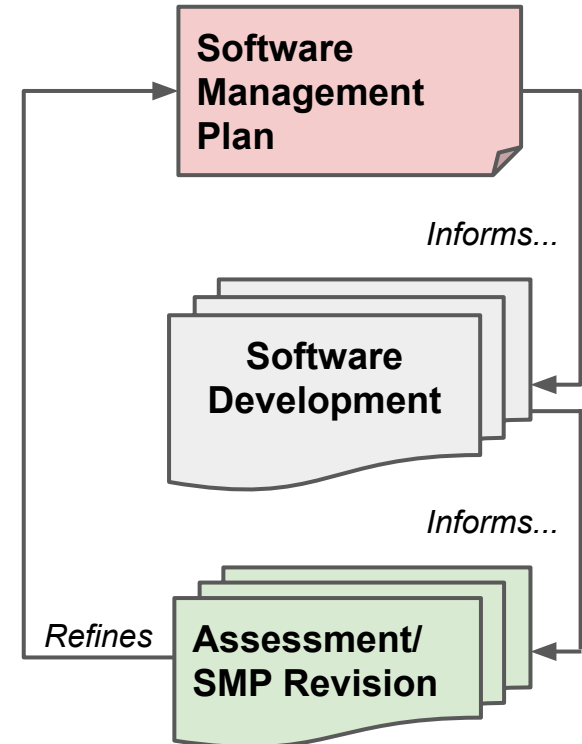
- Who is typically involved in making decisions?
- When is the right point to revise an SMP?
- How do you get hold of the info to put in an SMP?

# How to review SMPs?



www.software.ac.uk

- Need to know where you are
  - SMP is what you *intend*
  - Development is what is *actually happening*
- Involve stakeholders appropriately
- Cycle: assess, reconcile, refine
  - Again, prioritise



Software evaluation guidance & online Software Evaluation Service:

<https://www.software.ac.uk/resources/guides-everything/software-evaluation-guide>



- First steps - a draft set of SMP text and a plan to...
  - Put into a separate SMP document and build on
  - Discuss and agree with your team - *the true value*
  - Extend and evolve with your goals and project
- SMPs ideally concise, evolving documents
  - Once defined, set target dates for reviewing and revising
- Use the SSI's SMP checklist and guidance to help you
  - <https://software.ac.uk/software-management-plans>

# Reusing this material



www.software.ac.uk



This work is licensed under a Creative Commons Attribution 4.0 International License.

<https://creativecommons.org/licenses/by/4.0/>

This means you are free to copy and redistribute the material in any medium or format, and remix, transform, and build upon the material for any purpose, even commercially, under the following terms:

You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Acknowledge the Software Sustainability Institute as follows:

“© 2021 The University of Edinburgh, on behalf of the Software Sustainability Institute: [www.software.ac.uk](http://www.software.ac.uk)”