



Beyond Workshops

RSE-Led Hackathons as Experiential
Learning for Software Practice in the AI Era

Simon Stone, Will Cowen, Christian Darabos

Research Computing & Data @ ITC, Dartmouth College

Everyone's a software engineer now ...or are they?



Researchers and staff write more code than ever



AI assistants lower the barrier even further



The fundamentals have not changed: Git, tests, modular design, docs



Unguided AI yields working but unmaintainable code



The challenge: software that is reliable, reproducible and sustainable



How do we learn best?



Learn through experience

-  Workshops, tutorials and docs convey knowledge
-  Durable learning comes from hands-on work on real-life problems
-  Good Practice needs practice
-  The goal: create structured opportunities to practice



Image generated with GPT Image 2.5 Flare in Dartmouth Chat

Who we are

-  Research Computing & Data (RCD),
Dartmouth ITC
-  Sysadmins, facilitators, data scientists,
engineers
-  Shared infrastructure, but rarely shared
code
-  Silos limit knowledge transfer across
roles¹



Why should RSE lead?



We work with stakeholders at every technical level



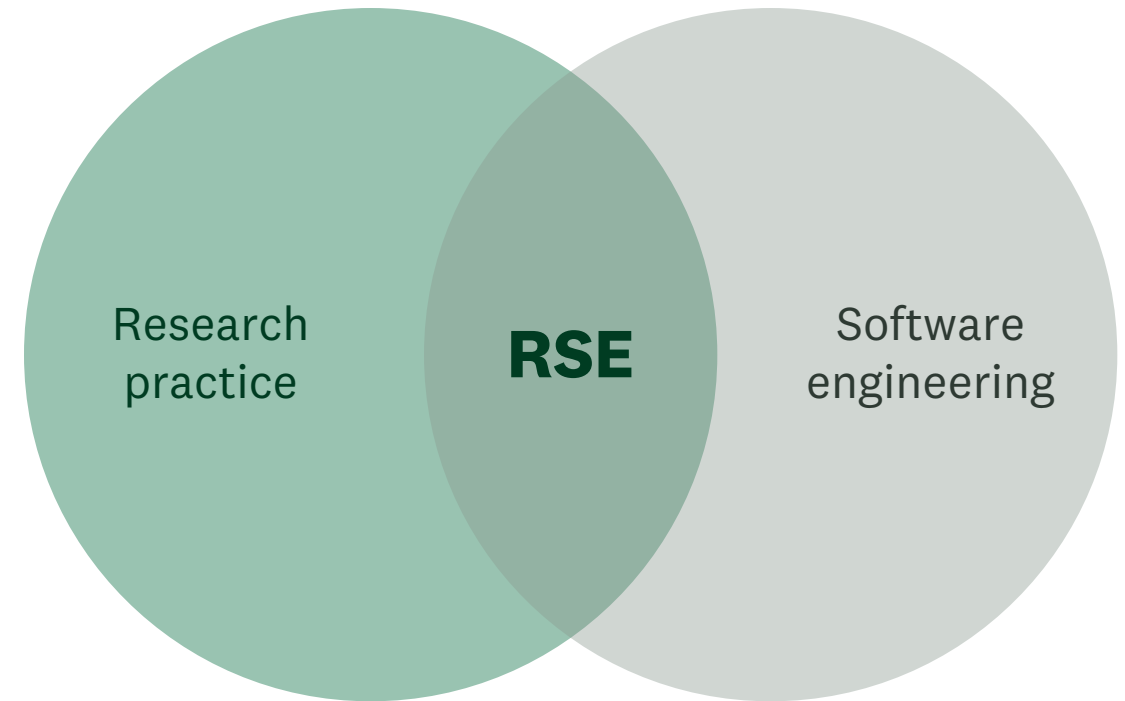
We translate research questions into software



We bring engineering discipline to research ambiguity

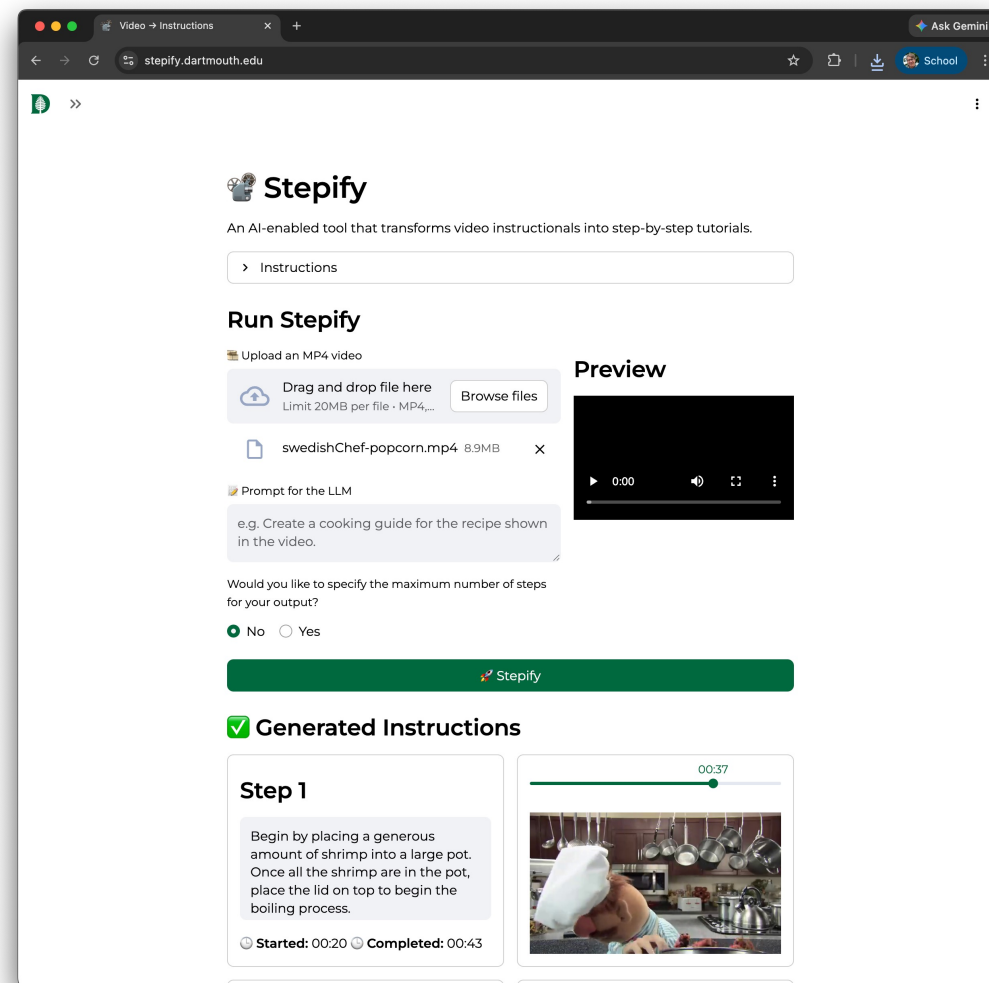


Three of the four RSE pillars:
development, community, training²



What we built

-  Stepify: AI video-to-tutorial converter
-  Videos in, Markdown guides with screenshots out
-  Frequent need: quick screen capture to impromptu tutorial
-  Python + Streamlit on the Dartmouth Chat LLM stack³
-  Open source and still in use
-  Product is not the point! (but we are proud of it)





How we got there



One day, 6 people from 4 teams



Started from a shared whiteboard sketch



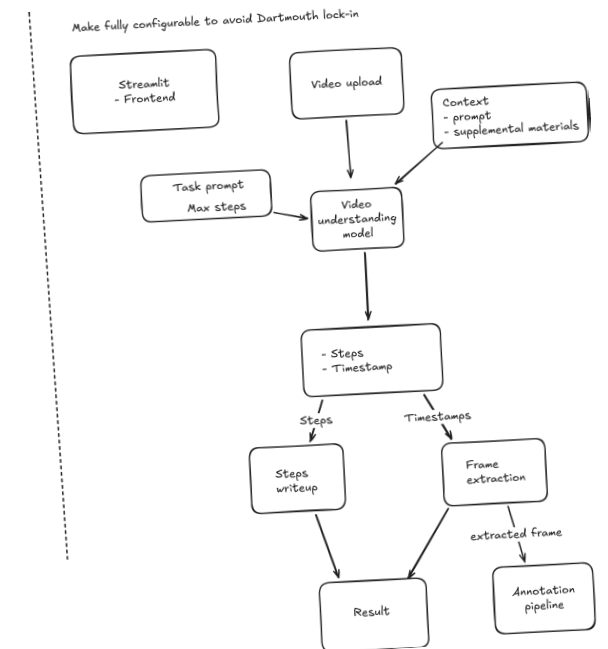
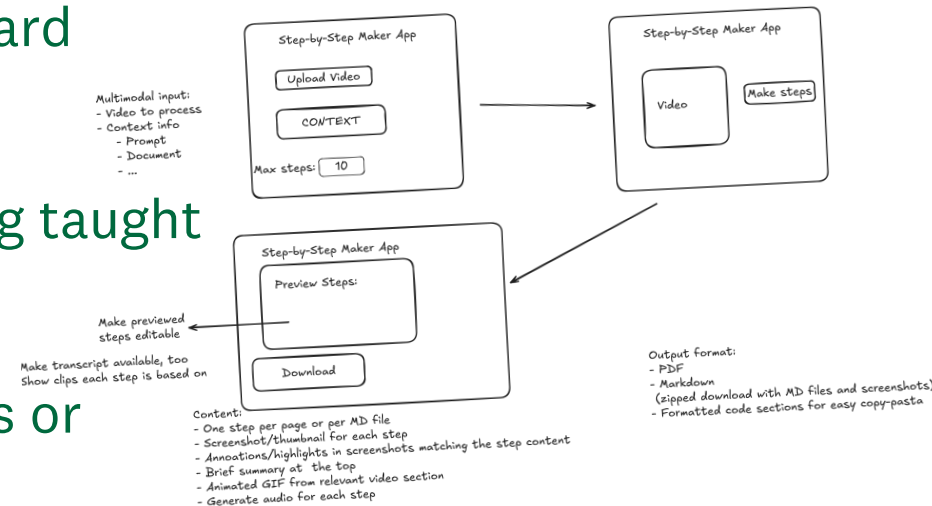
Git, task breakdown and testing taught in context



Existing LLM stack: no API keys or SDKs to manage



Working prototype by end of day



A job for every role

- 📋 Scope included non-coding work
- 🛠️ UI/UX sketches, test cases, evaluation, docs
- 🙋 Roles chosen by interest, not experience
- 🤝 Sysadmins and trainers worked alongside developers
- 🍯 Know your audience and come prepared with jobs you think they might like



Image generated with GPT Image 2.5 Flare in Dartmouth Chat

It's a sprint, not a race

 No prizes, no judges, no pressure⁴

Fixed time frame; focused collaboration

 Goal: a working prototype and shared learning

 Lower anxiety for less experienced staff

 Post-session clean-up/refactor

A prepared mind

 RSE prepared a preliminary architecture

 A starting point, not a specification⁴

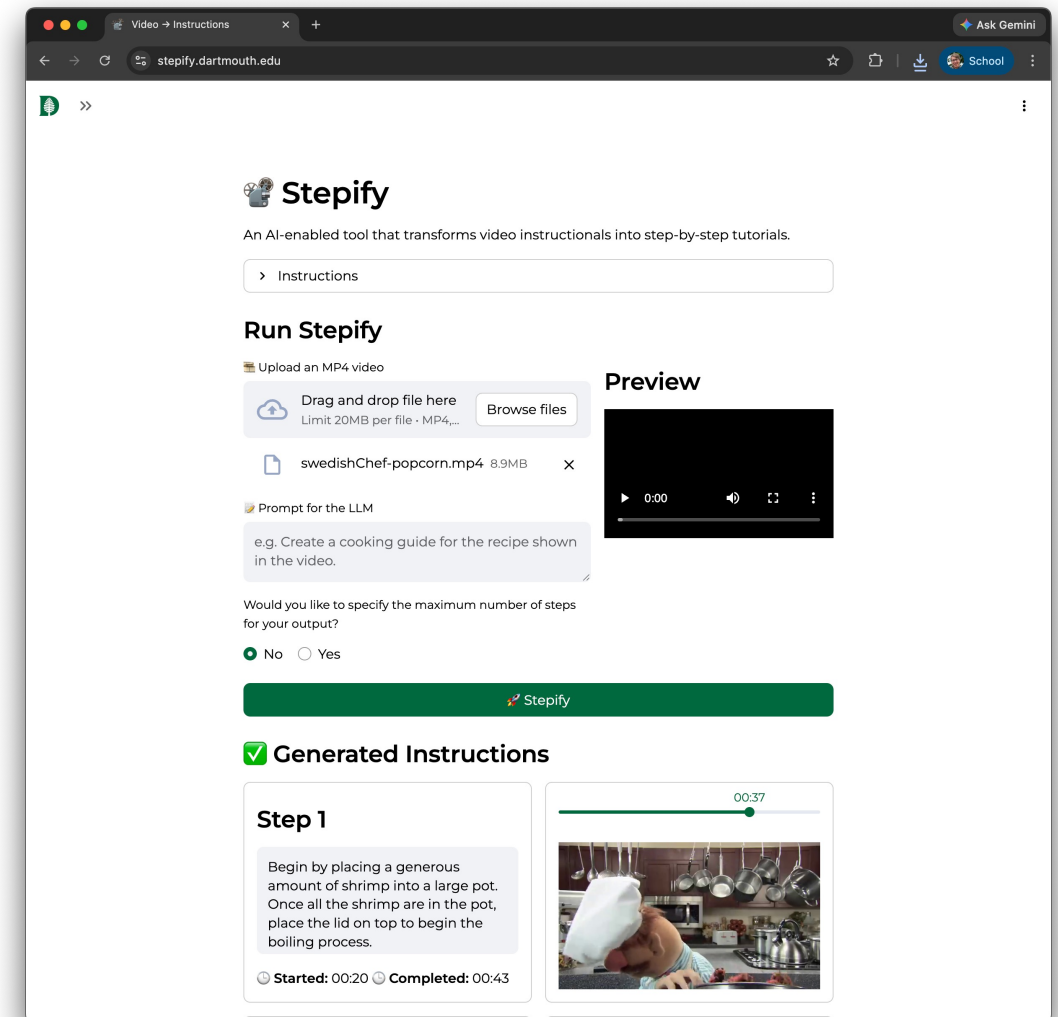
Skipped the “what should we build?” phase

 Others reshaped output format and annotation flow

 Keep an open mind, but grounded in what is feasible

What we got out of it

-  Working prototype by end of day, still in use
-  Staff who usually meet via tickets built side by side
-  Non-engineers more confident with dev workflows⁵
-  Follow-up interest in code review and joint projects



...but AI!!



Code is cheap to generate



People still decide what to build and why



Design, tests and review are more important than ever



The same skills direct AI agents: break down tasks, check outputs



Image generated with GPT Image 2.5 Flare in Dartmouth Chat



Run your own

- ✓ Pick an interesting/annoying/fun problem your team actually has
- ✓ Scope work for every role, not just coders
- ✓ Don't make it a competition, but make it a single day
- ✓ Bring a rough architecture, open to change
- ✓ Build on existing infrastructure to cut setup
- ✓ Teach in context



“People consistently surprise themselves when they are trusted to try.”

- a participant





Thank you

dartgo.org/usrse-beyond-workshops

Claude Opus 4.8 and 5.5 helped drafting this talk. Premise, outline, ideas, learnings, reflections, and mistakes are all ours.



References

1. A. A. Jones, J. Uhd, C. D. Kabore, K. A. Cornett. *Breaking down silos in the workplace: A framework to foster collaboration*. J. Public Health Management and Practice 30(6):E306–E311, 2024.
2. J. Cohen, D. S. Katz, M. Barker, N. Chue Hong, R. Haines, C. Jay. *The Four Pillars of Research Software Engineering*. IEEE Software 38(1):97–105, 2021. [doi:10.1109/ms.2020.2973362](https://doi.org/10.1109/ms.2020.2973362)
3. S. Stone, J. Crossett, T. Luker, L. Leligdon, W. Cowen, C. Darabos. *Dartmouth Chat – Deploying an Open-Source LLM Stack at Scale*. PEARC '25, Art. 43, 2025. [doi:10.1145/3708035.3736076](https://doi.org/10.1145/3708035.3736076)
4. A. O. Affia-Jomants, K. Gama, J. D. Herbsleb, A. Nolte. *How to organize an in-person, online or hybrid hackathon – A revised planning kit*. arXiv:2008.08025, 2020. [doi:10.48550/arXiv.2008.08025](https://doi.org/10.48550/arXiv.2008.08025)
5. E. P. P. Pe-Than, A. Nolte, A. Filippova, C. Bird, S. Scallen, J. D. Herbsleb. *Designing Corporate Hackathons With a Purpose: The Future of Software Development*. IEEE Software 36(1):15–22, 2019. [doi:10.1109/MS.2018.290110547](https://doi.org/10.1109/MS.2018.290110547)