SAND2020-1014C

IDEAS productivity
www.ideas-productivity.org

Lightweight Software Quality Practice

PSIP: Productivity Sustainability Improvement Planning

Elaine M. Raybourn, Reed Milewicz, Greg Watson, Elsa Gonsiorowski, Jim Willenbring
and the IDEAS-ECP team

ECP EXASCALE COMPUTING PROJECT

What is PSIP?

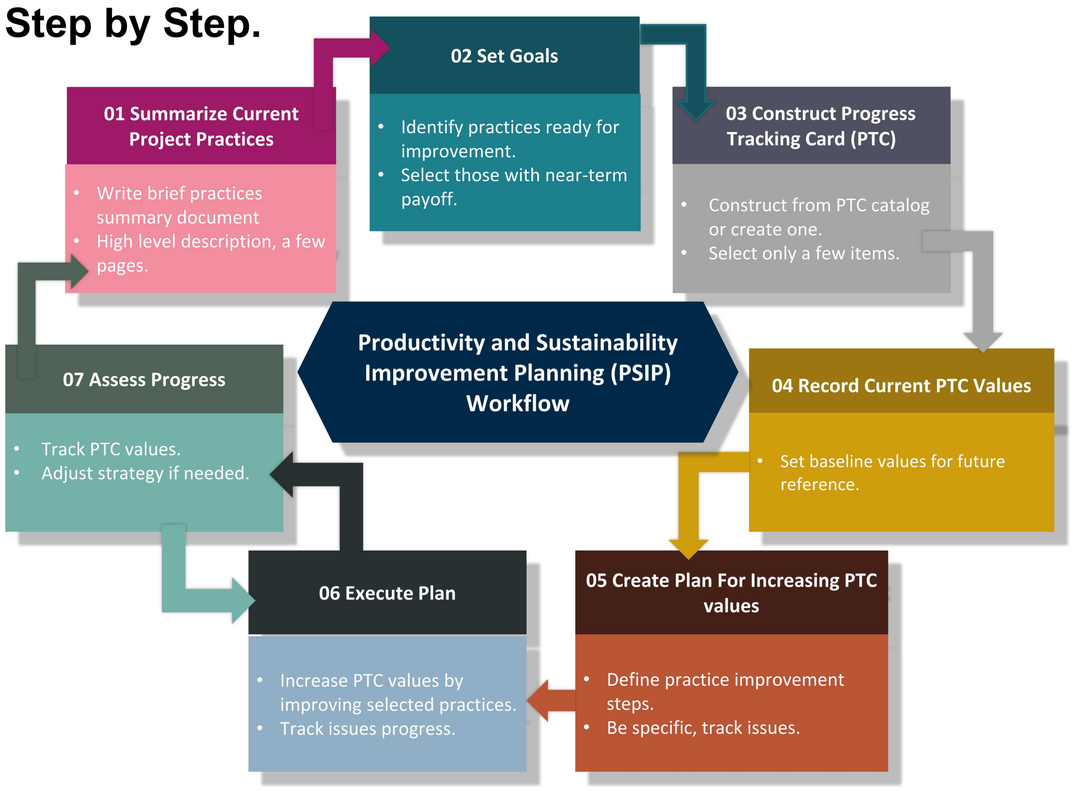
PSIP is a practice toward everyday software quality.

psip

Productivity and Sustainability Improvement Planning (PSIP) is a lightweight, iterative workflow that allows software development teams **IDENTIFY** opportunities to iteratively and incrementally **IMPROVE** software practices and processes [1]. By practicing PSIP, teams are able to **REALIZE** process improvements without a disruption to any current development processes. PSIP is a technique used alongside tools e.g. GitHub, Kanban, Agile, etc. or stand-alone.

How does PSIP help mitigate technical risk?

Step by Step.

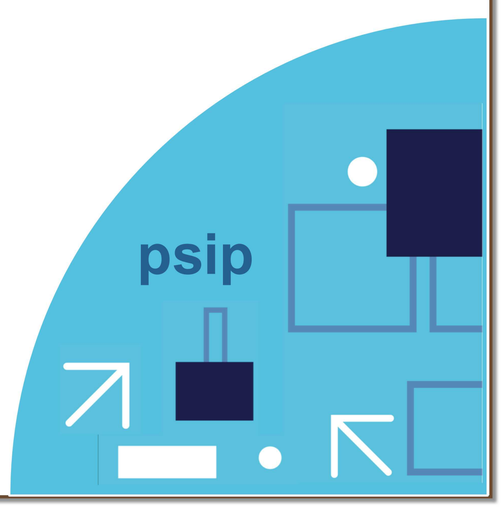


The key to PSIP is understanding where you are starting from, setting goals, and tracking your progress. The way you implement PSIP is by creating and using progress tracking cards (PTC). Tracking your progress can help mitigate technical risks to meeting project milestones.

Get started on your own or with a PSIP IDEAS-ECP facilitator. Group and team training available! We come to you.

Contact us bssw.io/psip

Upgrading team practices



Why Practice PSIP?

PSIP can help mitigate technical risk.

PSIP utilizes progress tracking cards (PTC). Progress tracking cards are concise visual aids that record your goals, deliverables, and milestones to internally compare your team's progress toward an intended goal. PTCs are not meant to be external assessments or evaluation tools. PSIP can help develop best practices and using PTCs will help keep track of progress toward an objective. You can also use or share PTCs when communicating to other teams about goals and how you know when you will have met your objectives.

When would I, or my team, initiate PSIP?

Use PSIP to upgrade team practices immediately.

PSIP can be initiated at anytime, especially when you or your team are 1) onboarding new team members unfamiliar with your current process, 2) your code has a new set of users/collaborators who need to understand your process for version control, and 3) the growth and success of your code base requires your team to interface with other teams more frequently. Practice PSIP to track software quality initiatives, reach goals, identify gaps, or upgrade team practices anytime.

Get started with the "Rate my Project" poster to your right.

How do we use Progress Tracking Cards?

Pull PTCs from a catalog, modify, or create your own.

We are populating a catalog of PTCs [2] so you can grab cards off the shelf or get started on your own. You can have more than one PTC, or several sub-tasks associated with PTCs. PTCs can be Agile Epics, GitHub Milestones, Projects, Issues, or even simple checklists. You and your team choose what meets your needs most. See our GitHub repo for examples on how you can integrate PTCs into your projects.

bssw.io/psip

Card: Testing	
0	**Initial Status :** No comprehensive testing framework in place. ✓
1	Add 1-3 example tests using the existing CMake infrastructure (CTest). ✓
2	Add 1-3 example tests using the 'Boost Test' library. ✓
3	Integrate the CTest infrastructure with the new Boost tests. ✓
4	Integrate the Boost-enabled CTest framework into the CI pipeline.
5	**Bonus:** Work with EXAALT team to add more advanced tests to improve code coverage.

PTCs created by EXAALT

Card: Continuous Integration	
0	**Initial Status :** No comprehensive CI framework in place. ✓
1	Develop a minimal Docker image with EXAALT dependencies. ✓
2	Implement a minimal 'yml' script for the CI pipeline. ✓
3	Update EXAALT Docker image to leverage CMake, and create a ParSplice-specific image for build testing. ✓
4	Generate step-by-step "how-to" Docker image documentation. ✓
5	Extend CI to automate build and functionality testing with both CMake and Boost. ✓

Practice: Test Coverage	Score (0 – 5):
Score	Descriptions
0	Little or no independent testing. Functional testing via users.
1	Independent functional testing of primary capabilities.
2	Primary functional testing, some unit test coverage.
3	Comprehensive unit testing, primary functional testing.
4	Comprehensive unit testing, functional testing for documented use cases.
5	Comprehensive unit, use case functional testing; test coverage commitment.

Example Card created by Mike Heroux

Comments:
1. **Functional testing:** Testing capabilities from user's perspective. Many functions can be called. Good for usability assurance. Insufficient to protect against some regressions. Difficult to isolate regressions. Can require extensive test execution times.
2. **Unit testing:** Isolated, independent testing of functions and methods. Enable test-driven development, rapid test execution, fault isolation. Insufficient to ensure functional correctness.
3. **Comprehensive:** Does not mean 100% line coverage, but sufficient coverage to detect most errors. Experts suggest various metrics such has 80% or more line coverage, or some similar high percentage of function point coverage.
4. **Commitment:** Team is committed to writing comprehensive tests concurrent with functionality.

Argonne NATIONAL LABORATORY
BERKELEY LAB
Lawrence Livermore National Laboratory
Los Alamos NATIONAL LABORATORY EST. 1943
OAK RIDGE National Laboratory
Sandia National Laboratories
UNIVERSITY OF OREGON

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2019-8403 C

[1] *Heroux M. Gonsiorowski, E., Gupta, R., Milewicz, R., Moulton, J.D., Watson, G., Willenbring, J., Zamora, R., & Raybourn, E.M. (2020-in press). "Lightweight Software Process Improvement using Productivity & Sustainability Improvement Planning (PSIP)", International Workshop on Software Engineering for HPC-Enabled Research (SE-HER 2019)*
[2] https://bssw.io/psip