



**PyPy: Researching a Highly Flexible and Modular
Language Platform
and
Implementing it by Leveraging the Open Source
Python Language and Community**



Project Goals

- Techniques to improve the implementation of dynamic languages:
 - Faster.
 - More easily customised.
 - More platforms.
 - More programming paradigms.

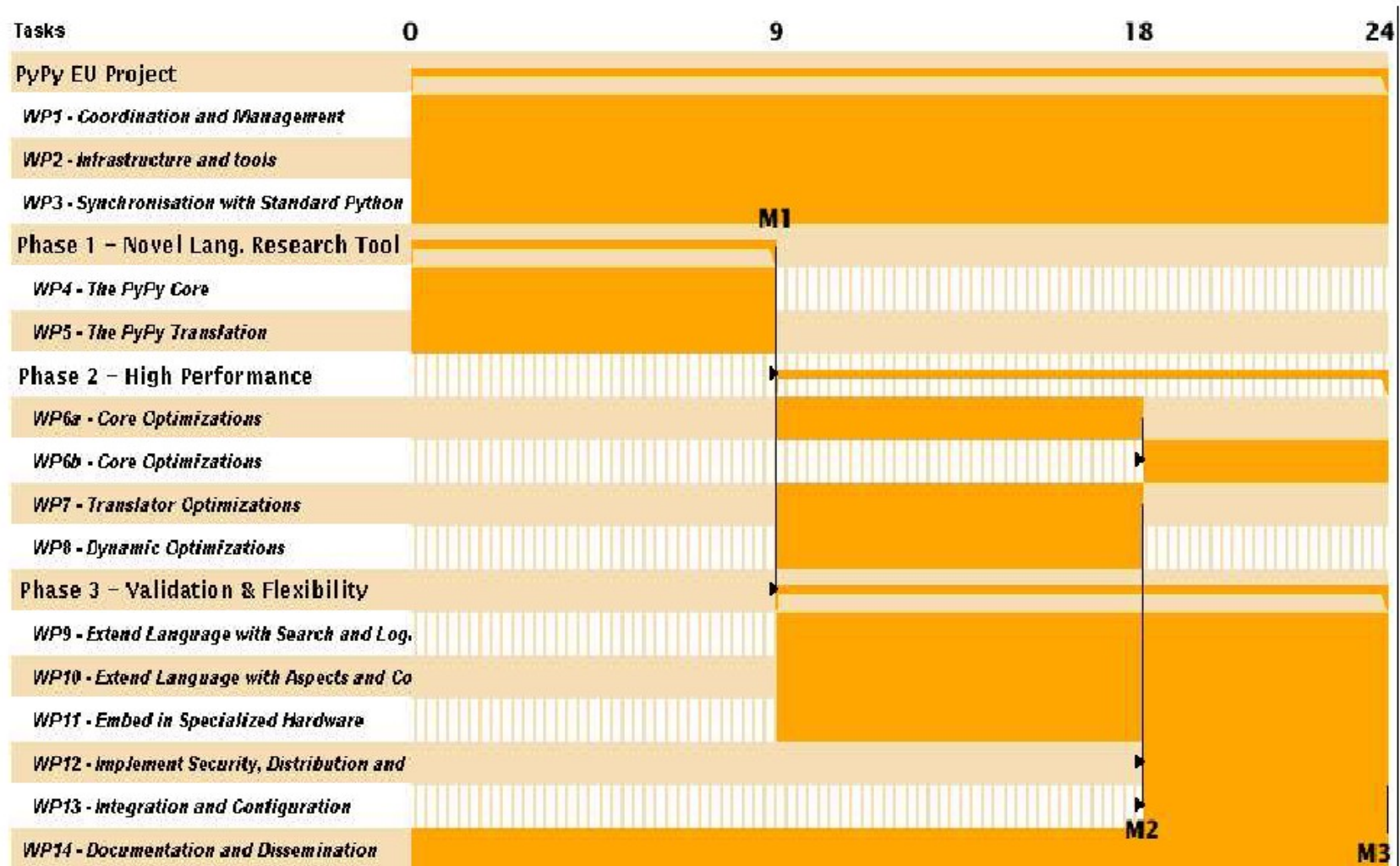


Concrete Goals

1. Python interpreter written in Python.
2. Analysis and translation of the PyPy core into efficient low-level code (C, Pyrex, Java, others).
3. JIT compilation techniques in Python.
4. A version of PyPy that supports real time massive parallelism.
5. Aspect-oriented programming in Python.
6. Search and constraint programming in Python.



Progress So Far





Code Development

- Typical Python development practices:
 - Write tests before code.
 - Typical tools: Subversion source-code management, IRC.
 - Regular programming workshops – *sprints*:
 - Ca. one week duration.
 - Focused on one topic.
 - Pair programming.
- Unique combination of EU and FOSS community project.



Implementation Techniques

- Novel ways to optimise a dynamic language:
 - Object spaces – building on abstract interpretation, modular VM.
 - JIT techniques.
 - Search constructs.
- Applicable to other dynamic languages.
- Search and Aspect Oriented Programming:
 - Optimisation problems, inference for the semantic web.