

EXAMENSARBETE User-Centric Study and Enhancement of Python Static Code Analysers**STUDENT** Steven Chen**HANDLEDARE** Emma Söderberg, Alan McCabe (LTH)**EXAMINATOR** Martin Höst (LTH)

Decoding Python: Making Code Analysis Tools Friendlier with User-Centered Design

POPULÄRVETENSKAPLIG SAMMANFATTNING **Steven Chen**

Imagine learning Python, but a tool keeps correcting your naming style. This is exactly what novice Python users often face when code analysers flags their variables. Our project combined user-centric research with practical enhancements to Pylint in Visual Studio Code, making Python's naming conventions less daunting.

In an increasingly digital world, programming has become a vital skill. But, for beginners, the experience can be daunting. Static code analysis tools - software that checks code for errors - are invaluable but often intimidating to the uninitiated. This is particularly true for Python, a popular programming language, where the needs of novice users have been largely overlooked.

While Python analysers tools are invaluable for maintaining code quality, they still present many usability challenges that hinder beginners from using them. Addressing this, our degree project adopted a dual approach: insightful user-centric research followed by a practical enhancement of Pylint, a static code analysis tool designed to improve Python code.

First, we dived into the world of novice Python programmers by conducting interviews and surveys. This helped us understand their struggles and frustrations. Armed with these insights, we shifted to the next phase: enhancing Pylint in Visual Studio Code by providing additional quick-fix options for the 'invalid-name' issue.

Instead of forcing users to rectify naming style errors immediately, these quick-fixes offer them

the option to ignore the naming rule at different levels. This introduces an element of flexibility to the learning process, allowing beginners to focus more on understanding Python's basics and less on adhering to strict naming conventions.

The significance of our project lies in its potential to make the early steps of learning Python less intimidating. Code analysis tools like Pylint are integral to writing high-quality code. By making this tool more user-friendly, we create a more inclusive coding environment.

Considering the future implications, our enhancements to Pylint aim to improve the usability of Python analysers overall. By making these tools more accessible and user-friendly, we're not only smoothing the initial journey for beginners but also creating a more efficient coding environment for all users.

In sum, our project exemplifies the fusion of technology and user-centric design. It underscores the importance of user feedback and empathy in creating tools that better meet user needs. Ultimately, we hope our work will serve as a step forward in making Python static code analysers more accessible and user-friendly.