To put static program analysis at the fingertips of the software developer, we propose a framework for interactive abstract interpretation. While providing sound analysis results, abstract interpretation in general can be quite costly. To achieve quick response times, we incrementalize the analysis infrastructure, including postprocessing, without necessitating any modifications to the analysis specifications themselves. We rely on the local generic fixpoint engine *MegaFix*, which dynamically tracks dependencies, while exploring the unknowns contributing to answering an initial query. Lazy invalidation is employed for analysis results affected by program change. Dedicated improvements support the incremental analysis of concurrency deficiencies such as data-races. The framework has been implemented for multithreaded C within the static analyzer *WizWoz*, using *WizWozBridge* to relay findings to IDEs. We evaluate our implementation w.r.t. the yardsticks of response time and consistency: formerly proven invariants should be retained – when they are not affected by the change. The results indicate that with our approach, a reanalysis after small changes only takes a fraction of from-scratch analysis time, while most of the precision is retained. We also provide examples of program development highlighting the usability of the overall approach.