**The overall strategy of the work plan**

The FITTEST project is divided into 11 WPs as indicated in Figure 1.3.1.

As in any effective project there is a need for management, dissemination and exploitation which will be covered by WP1 and WP11 respectively. WP11 shows how the FITTEST project has an outstanding and imaginative plan for exploitation and dissemination that will ensure maximum impact of the results. WP1 shows that the whole project is underpinned by a carefully planned management program, which has been „battle tested“ and proved to be exceptionally effective in the highly successful EvoTest project.

The goal of WP3 is to lay the foundations for the FITTEST project by defining and formulating the objectives and fitness functions required for optimization tasks in WP4-WP8. This part of WP3 will be laid using well understood techniques and will constitute a robust and stable foundation upon which to build. WP3 will also seek to innovate and develop imaginative new approaches to search based optimization of future internet testing, using genetic programming and co-evolutionary optimization.

WP4 addresses the highly dynamic nature and the low observability of FI models by combining dynamic analysis and search based algorithms to develop a dynamic analysis framework for the automated inference of FI testing models, to be used for automated test case generation, in order to address directly and explicitly the challenges of self-modification and autonomic behaviour. Dynamic analysis for model inference coupled with continuous testing provides a mechanism to test a self-modifying system with minimal human intervention. The under-approximation, intrinsic of dynamic analysis, will be mitigated by resorting to search based techniques from WP3, which will be oriented toward maximizing the level of state exploration achieved by the inferred model. A UML2 Profile will be defined for the commercial tools of SOFT.

In WP5 the Classification Tree Method (CTM) [GG93] for combinatorial testing will be extended. For FI technologies, the number of classifications could become very large, resulting in a tremendous amount of possible test cases. In order to reduce this huge amount, trees will be trained to partition the relevant configuration space into equivalence classes, based on the behaviours and application states observed at run time, during monitored executions. This allows us to apply combinatorial testing techniques even in the presence of extreme dynamism and self-modifications. Search techniques will be used to find a minimal set of test cases fulfilling the "combination requirements" introduced by the tester.

WP6 will develop a diagnosis approach that will allow us to find vulnerabilities in FI
applications by analysing co-generated logs. Moreover, oracle will be automatically generated through learning techniques such that the prediction of expected values to determine whether a test has passed or failed can be done automatically.

**WP7** will develop a concurrency testing and debugging environment for FI applications that specifically supports easy manipulation and reproduction of the conditions under which concurrency bugs appear. Combined with search based test case generation from WP3, this approach has the potential to make a major contribution to a fully automated, continuous search for intricate and complex concurrent bugs.

**WP8** will work on techniques that give useful coverage feedback on FI applications. Since FI applications are huge, even in good testing situations, only low coverage percentage may be achieved. WP8 will investigate a qualitative interpretation of coverage that goes beyond the quantification of the coverage percentage. We will take advantage of linguistic, informal information associated with code elements to qualify coverage in terms of high level, human understandable functionalities that are indeed covered or yet to be covered.

**WP9** deliver a common Integrated Testing Environment where resulting tools from WP3-WP8 will be integrated according to the architecture displayed in Figure 1.1.1. The resulting environment will be for the automated testing of FI applications that will be evaluated by the case studies in WP10.

In order to assess the FITTEST results, the partners will evaluate the project with respect to the now well-established principles of Empirical Software Engineering. This will be addressed in **WP2**, which will formalise and develop techniques for detailed assessment of the performance of the testing techniques developed. The results of these empirical evaluations will make it possible to answer fundamental questions about the effectiveness, efficiency, applicability and reliability of FITTEST results for FI.

**WP10** will evaluate the FITTEST results on real world FI applications that have indicated to have testing problems related to the challenges identified in Section 1.1 and will be solved through the FITTEST objectives. WP10 ensures that the project achieves this necessary real world application of testing FI systems in order to achieve maximum impact within the industrial sector. Moreover, WP10 makes that the project will maintain, throughout, a focus on real-world relevance and exploitability of the results.