Avoiding Software Test Automation Failure

Introduction
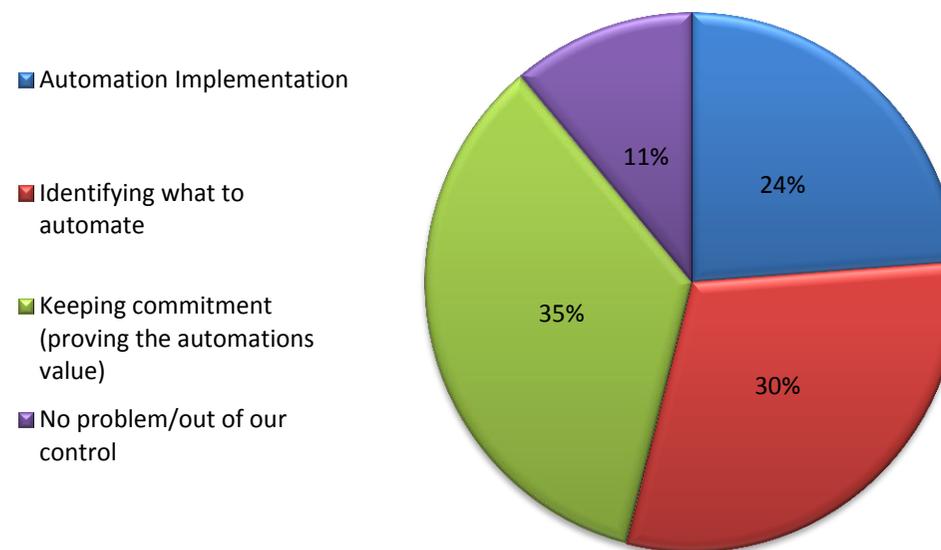
To continue to improve our knowledge of our core market challenges, CodeFuse Technology regularly conducts research into the problems software developers and quality professional's face.

This paper contains analysis of the reasons why test automation projects fail. We hope that this will provide useful information to any readers looking to start their own automation projects.

This research includes input from over 50 software professionals captured from both face to face meetings and online discussions.
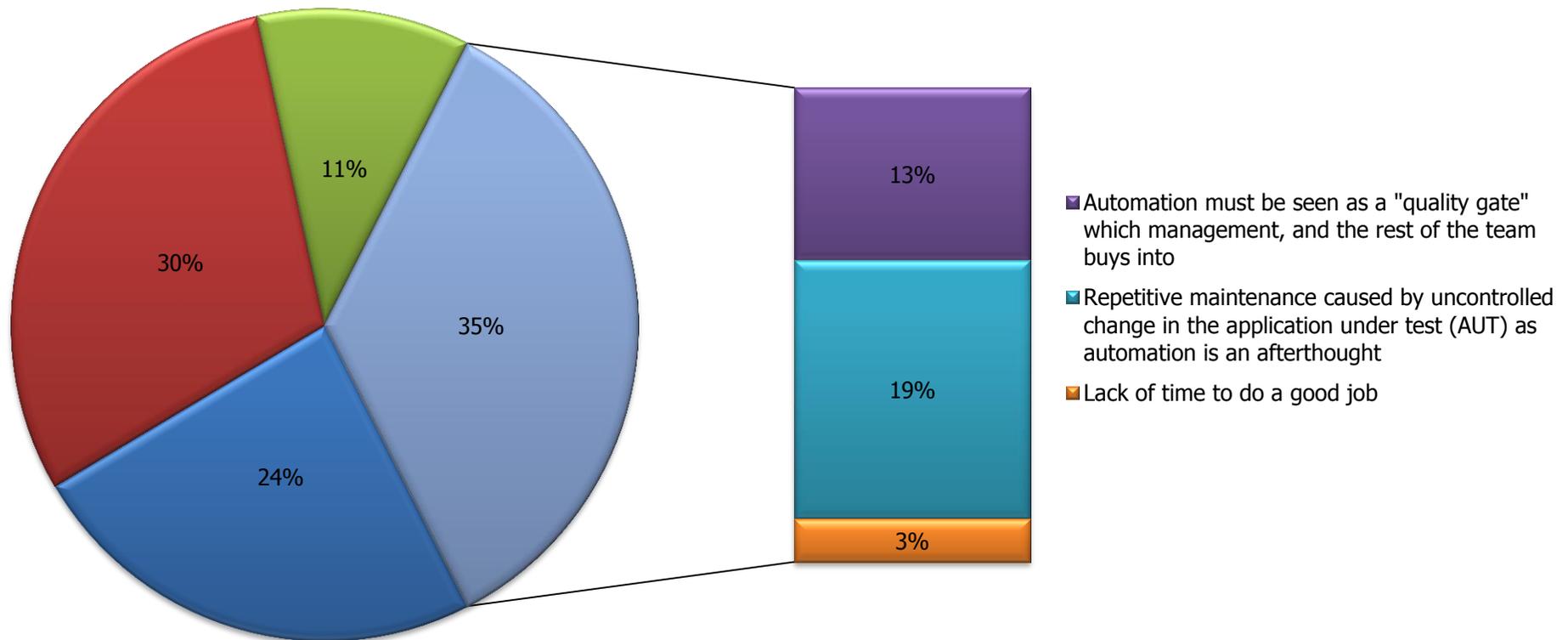
Findings and Analysis

To summarise our findings, we found that the reasons for failure could be logically grouped under four core areas as shown in the chart below:
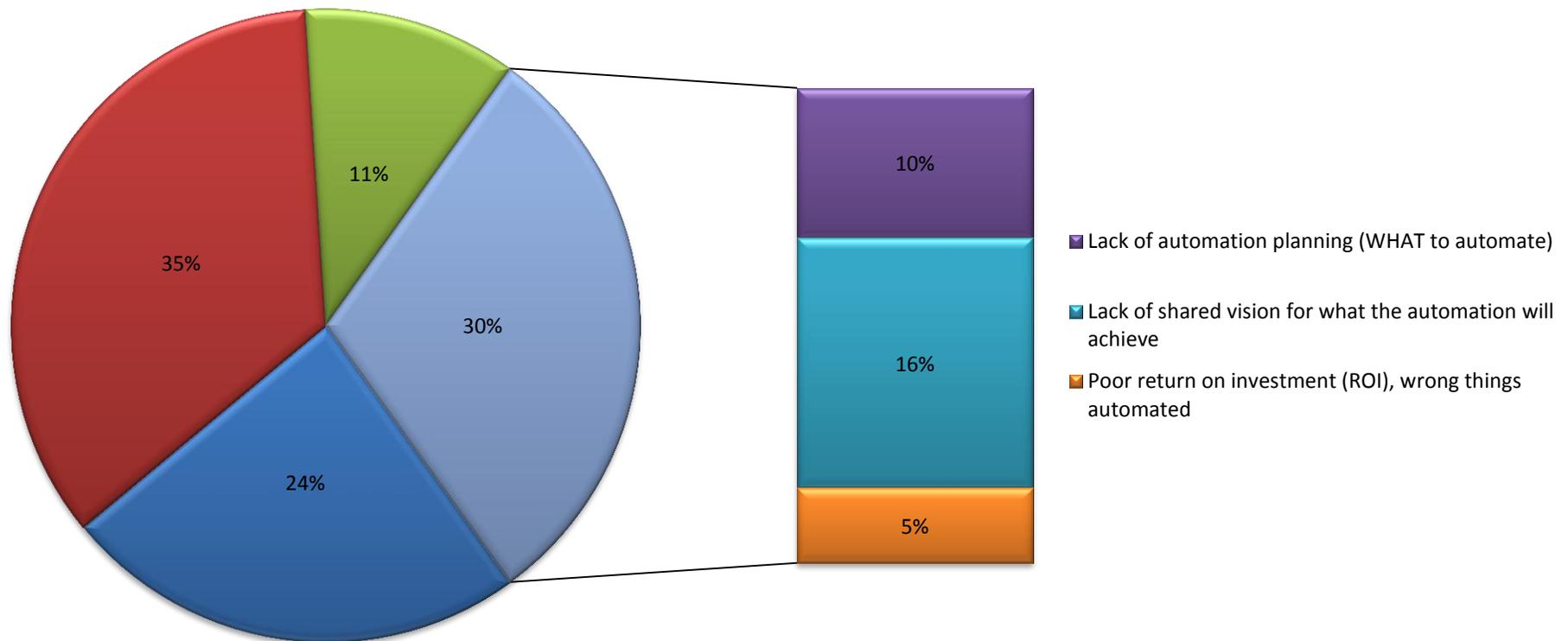

- Automation Implementation
- Identifying what to automate
- Keeping commitment (proving the automations value)
- No problem/out of our control

The largest of these groups was "keeping commitment (proving the automation's value)", which was broken down as follows:

Keeping commitment (proving the automation's value)

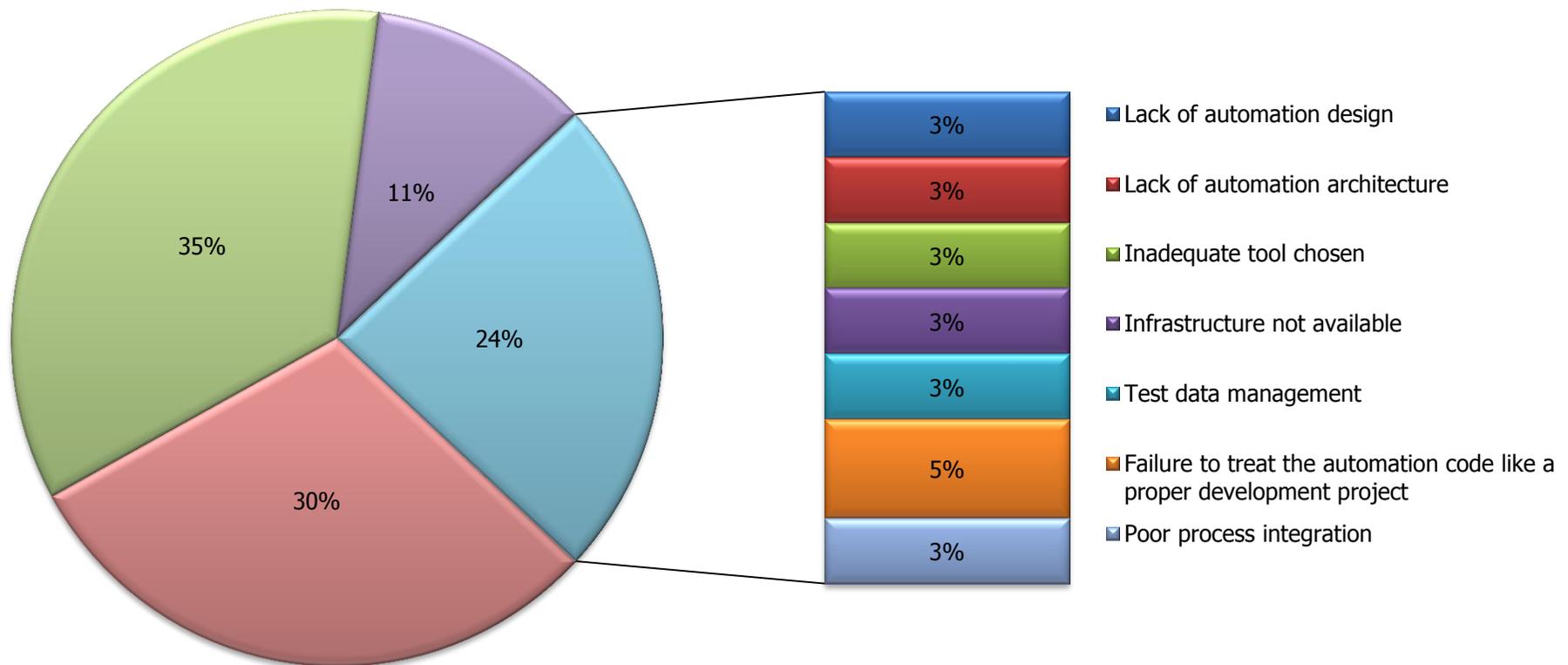- Automation must be seen as a "quality gate" which management, and the rest of the team buys into
- Repetitive maintenance caused by uncontrolled change in the application under test (AUT) as automation is an afterthought
- Lack of time to do a good job

The next largest group was "identifying what to automate":

Identifying what to automate

11%

35%

30%

24%

10%

16%

5%

- ■ Lack of automation planning (WHAT to automate)
- ■ Lack of shared vision for what the automation will achieve
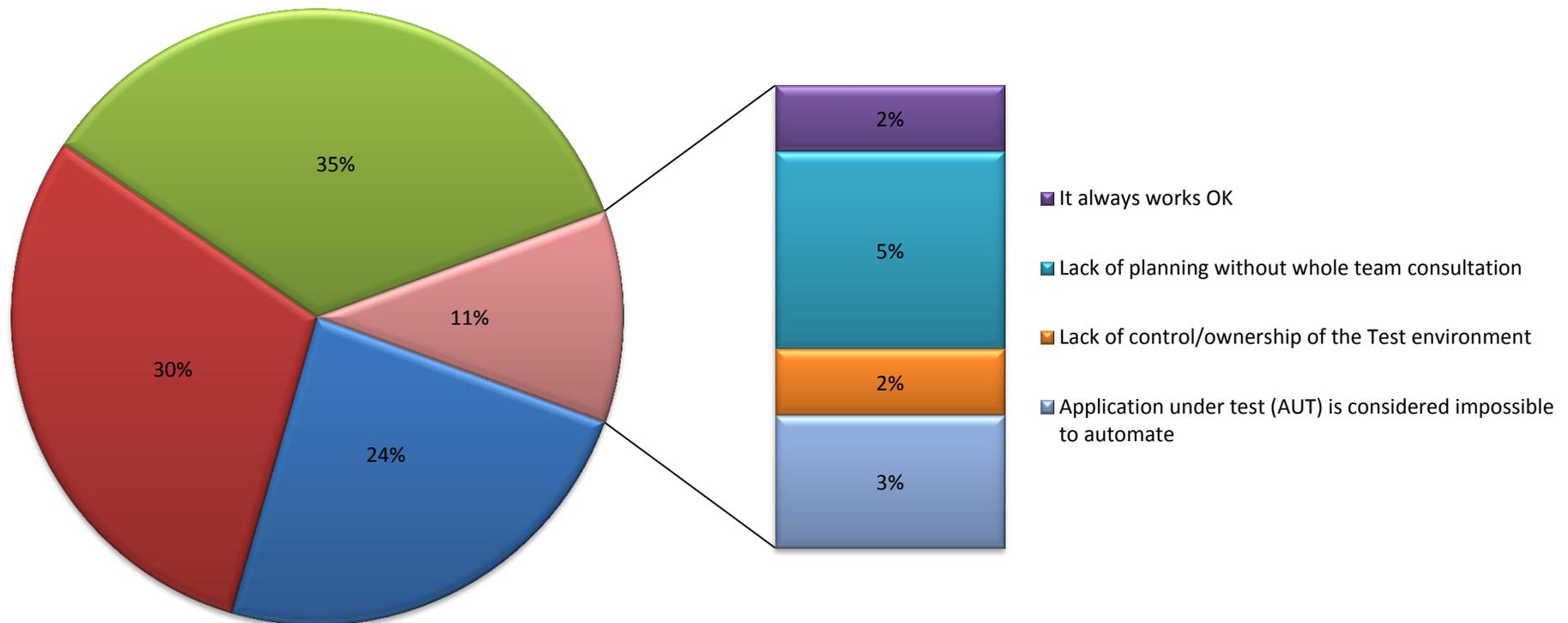- ■ Poor return on investment (ROI), wrong things automated

And the last of the main groups, although still considerable in size, was "automation implementation":

Automation Implementation

3%	■ Lack of automation design
3%	■ Lack of automation architecture
3%	■ Inadequate tool chosen
3%	■ Infrastructure not available
3%	■ Test data management
5%	■ Failure to treat the automation code like a proper development project
3%	■ Poor process integration

Finally, some of the responses we received were either that the person had never experienced a problem, or the problems were perceived to be beyond the control of the development team.

No problem/out of our control

35%

30%

24%

11%

2% It always works OK

5% Lack of planning without whole team consultation

2% Lack of control/ownership of the Test environment

3% Application under test (AUT) is considered impossible to automate

Conclusion

While most teams will invest time in choosing a tool which is fit for purpose, our research suggests that in addition they need to look at all three core areas outlined in this paper.

Choosing a tool is important but needs to be accompanied by a well thought out architecture for your automation, a way of storing test data and management of devices to run tests on as well as a number of other factors.

While the technical side is important, it is clear that getting your team, or possibly the whole department; to share a common vision is paramount. In getting consensus on what the test automation vision is; articulating what you are going to automate is vital.

Finally, because quality is often an area which is difficult to quantify and open to interpretation, you need to keep team commitment high. Key to success is strong communication of the value of your automation and the benefits it brings.

CodeFuse Technology Solutions

CodeFuse Technology are to start beta testing two products in 2015 which address many of the issues in the three main failure areas for test automation. CodeFuse Console addresses the area of "automation implementation". CodeFuse QualityMap tackles the issues associated with "identifying what to automate" and "keeping commitment through proving value in test automation".

See our website at http://www.codefuse.io/ for more details.