

# WORKRAVE TEST STRATEGY

This strategy is the result of a TestLab session at EuroSTAR 2014 in Dublin. The situation is fake, but the software is available and free for download at [workrave.org](http://workrave.org).

A handful of people used James Bach's Heuristic Test Strategy Model, to get information about what is important. This document was constructed afterwards by Rikard Edgren, and pretty much sums up the ideas we had during the session.

## Introduction

For the first time, WorkRave will be tested by testing professionals. Christopher will be the tester, but will make use of other people to get a broader coverage. There is no hard deadline, so Christopher will decide on the details, together with the product manager, especially where to put most focus is open for discussion.

## Testing Missions

The testing missions as understood by talking to our proxy, Ruud are:

1. Find important problems, especially around breaks, UI consistency and perceived charisma
2. Compare WorkRave with competitors
3. Identify valuable enhancements
4. Find information/evidence that shows that WorkRave does what it is supposed to do

We explicitly don't need to find more information to add to user manuals.

## Information Sources

Since there are no requirements, we will mainly use information from conversations with developer and product manager, together with [workrave.org](http://workrave.org) and competitor information.

## Quality Objectives

We believe that these are the intended quality objectives, but want input from other people!

**Capabilities** Accurate reminders (within 1 second) for micro-break, break and daily limit.  
Correct monitoring (2 methods) of computer activity (also for special devices)  
Unclear if Network feature is central or not.

**Reliability** Always remind, also under stress.  
Informative error messages.

**Usability** Easy to start and get going.  
Easy to understand how to tweak limits to your convenience.  
Understandable visual and audio output (might there be blind users?)  
Information about why the software helps for RSI

**Charisma** help to take breaks, without being (too) annoying  
invisible when breaks are respected

**Security** It should not be possible to hack into a company via WorkRave (Network feature)

**Performance** No significant delays when using the software  
Minimal resource usage, that doesn't increase over time

**IT-bility** Easy to roll-out in a big company  
User settings should be retained after upgrade

**Compatibility** same behavior on Windows 7, Windows 8 and Linux (new version, lower priority)

**Supportability** log files for overview and troubleshooting

**Testability** observability of current time status

**Localization** Exercises must be very carefully translated

## Christopher's Activities

Christopher will do most of the testing, aiming towards the relevant quality objective listed above, especially covering areas like:

- Run WorkRave and competitors on the same machine, to evaluate correctness, usability and charisma
- Thorough testing of break timings
- Fast testing of other functionality
- Test both versions of monitoring
- UI Consistency
- Clear terminology
- Have sound turned on, it is part of the charisma
- Resource usage (look in Task Manager from time to time)
- Use WorkRave for a long time period
- Try to provoke failures, to test reliability and error handling
- Security penetration testing?

Windows 8 will be the main platform (since developer uses Windows 7), but now and then, we will do some of the testing on Windows 7 and Linux (recent version.) User will be "restricted" in a way that we expect larger companies to use.

Christopher will also read documentation on competitors' web sites, especially looking for information/evidence about why the breaks are good for RSI prevention and recovery.

## User testing

In order to evaluate how good WorkRave is compared to competitors, we will do user testing, where a group of people will try both WorkRave and competitors. The main focus is to get inter-subjective answers to questions like:

- Which one do you prefer, and why?
- What do you like about WorkRave?
- What do you dislike about WorkRave?
- Was something hard to understand how to do?
- What was your first impression of WorkRave?

This will hopefully cover usability and charisma aspects, and give us good ideas about how to get better than competitors.

It would be very useful if we could let persons suffering from RSI to try WorkRave, to see if it actually helps.

Another idea is to get tester friends to try out WorkRave, and give their input.

## Physiotherapists

We should get in touch with physiotherapists that can give input on how good WorkRave is. Especially the default time settings need to be carefully selected. They can also help us by pointing to resources that show why these breaks help, and what to think about with time settings.

The exercises need to be good according to physiotherapists, and also easy to understand for a broad user base. It is desirable, but unrealistic (?) to have every exercise localization reviewed by a physiotherapist with good knowledge of the language.

## Challenges

If “Network” is an important area, it might take time to setup an environment that is similar to what companies will use.

If “Network” is important we should do security testing, that we don’t know if Christopher is capable of.

It will be difficult and/or expensive to get all translated exercises reviewed.

One success factor is the product’s charisma, which is quite subjective and could be challenging to report in a good way.

Rumour says that there is an API that might be very useful to make tests for reliability, endurance and more. We don’t know how this works, and if it is difficult to make use of.

## Reporting

There will be weekly status meetings, bug reports will be reported continuously, and a test report will be written in the end, containing the test coverage, and the results for each of the four testing missions.