



Robotic Process Automation Software

# Blue Prism

## TEST PHASES OVERVIEW

Version: ##

For more information please contact:

info@blueprism.com | UK: +44 (0) 870 879 3000 | US: +1 888 7577476

[www.blueprism.com](http://www.blueprism.com)

## Contents

1. Introduction .....	3
1.1. Blue Prism's Robotic Automation.....	3
2. Test Phases – Purpose of this Document .....	4
3. Object Testing .....	5
4. Component Testing and Test Rigs.....	6
5. Process Testing .....	7
5.1. Process Integration Testing.....	7
5.2. Configuration Testing.....	7
6. Verification.....	8
6.1. Pre-Verification Walk Through.....	8
6.2. Verification.....	8
7. User Acceptance Testing (UAT).....	9
8. Live Proving.....	10
9. Implementation Testing.....	11
10. Regression Testing .....	12

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third party without the written consent of an authorised Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited

© Blue Prism Limited, 2001 - 2015

All trademarks are hereby acknowledged and are used to the benefit of their respective owners.  
Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, Centrix House, Crow Lane East, Newton-le-Willows, WA12 9UY, United Kingdom  
Registered in England: Reg. No. 4260035. Tel: +44 870 879 3000. Web: [www.blueprism.com](http://www.blueprism.com)

## 1. Introduction

### 1.1. Blue Prism's Robotic Automation

Robotic Automation refers to process automations where computer software drives existing enterprise application software in the same way that a user does. This means that unlike traditional application software, Robotic Automation is a tool or platform that operates and orchestrates other application software through the existing application's user interface and in this sense is not "integrated".

Blue Prism's Robotic Automation software enables business operations to be agile and cost-effective through rapid automation of manual, rules-based, back office administrative processes, reducing cost and improving accuracy by creating a "virtual workforce".

The virtual workforce is built by the operational teams or accredited Blue Prism partners using our robotic automation technology to rapidly build and deploy automations through leveraging the presentation layer of existing enterprise applications. The automations are configured and managed within an IT-governed framework and operating model which has been iteratively developed through numerous large scale and complex deployments.

## 2. Test Phases – Purpose of this Document

This Test Phases document describes the standard test phases during a Blue Prism project to ensure that automated solutions are delivered into live with the optimum possible level of testing throughout development to ensure that processes are delivered that meet business requirements and contain the minimum possible levels of system exceptions.

As outlined in this document, testing is carried out through each stage of the Blue Prism methodology, through the creation of objects, components, processes, verification, and user acceptance testing. Experience has found that commencing robust testing as early as possible in the creation of a solution assists in finding and incorporating process scenarios and system behaviour that may otherwise have been missed during design and configuration.

### 3. Object Testing

As applications are modelled and actions are created in object studio they should always be tested as they are being developed.

The following steps should be taken during object testing:

- As each application element is modelled the 'highlight' facility should always be used to ensure it has been identified correctly and will be able to be used
- Actions should be step tested using the debug tool as they are built, and as each application element is used
- Running actions end-to-end in object studio at the fastest debugger speed will ensure adequate waits are built into the actions
- Running related actions in order, one after another, will ensure that they will work together as expected when used
- Actions should be tested with different inputs (i.e. different account numbers)
- Where test environments are not available break points and a 'test mode' should be built into update actions to ensure that no updates are done on live data until the verification phase.

## 4. Component Testing and Test Rigs

Once all the actions are created that are required for a task a component can be created. A component contains multiple completed object studio actions linked together (for example all the navigation and read and write actions required to retrieve client details). Components should include exception handling to retry if there is an unexpected system exception.

The following steps should be taken during component testing:

- A test rig should be created for each component. This is a small test process that runs the component multiple times for different accounts.
- Component test rigs should be ran in Control Room. This is the earliest opportunity to run sections of the solution at full speed.
- Test rigs should be ran with memory logging turned on. The logs should be checked to ensure that there are no memory issues (i.e. memory leaks) in Blue Prism or the application it is interfacing with.
- If no test environment is available for the system being used, components should be run in 'test mode' to ensure that no updates are done.
- Whilst testing components in control room, the system being used should be manually closed and/or navigated to incorrect screens so that the exception handling within components can be tested. Components should be able to return to a 'known location' with the application and continue.

## 5. Process Testing

### 5.1. Process Integration Testing

The first opportunity to test that components will work as expected when used in succession is as a process is being built in Process Studio. It is therefore recommended that processes are tested end-to-end as early as possible, even as it is being built and before all the business logic has been added.

The following steps should be taken during Process Integration testing:

- As multiple components are added to the process they should be step tested using the debugger tool, stepping into and out of each component. This will ensure components 'hang' together and that all the input and output parameters of each component are used correctly
- If there is no test environment available for an application, break points and 'test mode' should be used to ensure that no updates are done in live
- As each new scenario is added to the process it should be tested. A method of keeping track of which scenarios have been tested should be used (either a separate record, or MI built into the process)

### 5.2. Configuration Testing

Process testing ensures that the process works as outlined in process and solution design documents. The developer should ensure that all the specified functionality is built into the process and it works as expected, and the process is ready for verification.

The following steps should be taken during configuration testing:

- A list of the required process functionality should exist, created during solution design. This can either be in the form of a Configuration Test Plan, or detailed within a Scenario Tracker. Configuration testing should ensure that all the required functions have been built and are incorporated into the process.
- The process should be tested in both Process Studio and Control Room. The developer should be confident prior to verification that the process can be stepped through end-to-end
- If no test environment exists for the systems used, 'test mode' should be turned on

## 6. Verification

Verification is a vital phase in the Blue Prism methodology that ensures that the delivered process meet requirements, all known scenarios have been recognised and tested, and an opportunity is given to the business to identify anything previously missed in the process documentation.

### 6.1. Pre-Verification Walk Through

During the pre-verification walk through the process is quickly validated by a Test Analyst (TA), or Subject Matter Expert (SME), end-to-end by sight, giving the opportunity to quickly identify any obvious omissions or changes required.

- The developer explains the Blue Prism process to a TA/SME, 'walking' through the process diagram that has been built in Blue Prism with reference to process documentation.
- The TAs/SMEs are familiarised with the Blue Prism process in preparation for the verification phase. A brief overview of the Blue Prism product and process is given in preparation for assisting in verification, and a brief overview of how the product works is given (i.e. how components are stepped into, how work queues store information etc)

### 6.2. Verification

The verification phase involves stepping through the completed Blue Prism process with TA's/SMEs:

- Cases are stepped through in Process Studio with a TA/SME. Where possible TAs/SMEs should be experienced users that know the process well.
- As many different cases are used as possible to identify and test different scenarios
- As many different scenarios as possible should be tested. A Verification Test Plan or a Scenario Tracker can be used to ensure as many scenarios as possible are known.
- If any scenarios are not seen during verification testing (because a relevant case is not found) should be noted. A business decision will need to be made on each untested scenario to decide if it should be 'turned off' until it has been tested.
- The developer should force the process down routes in the process for which test cases have not travelled (either the route is purely theoretical or a test case for that route or scenario has not been found).
- Different SMEs should be used during the verification phase as different users will have different process knowledge.
- The verification phase should be re-iterative. Required changes identified with the help of the TA/SME should be developed, and the process re-verified.
- If no test environment is available, the verification phase is the first opportunity to test the performing of updates (i.e. clicking confirmation buttons). The TA/SME should confirm that Blue Prism is navigating to and completing update screens correctly before allowing Blue Prism to perform the updates. Any messages or pop-ups not previously seen will need to be configured.

- **Note: If test systems are not available and access to the live systems are not made available for verification, additional time will need to be factored in to UAT and live roll out of the solution**

## 7. User Acceptance Testing (UAT)

UAT is a customer driven testing phase, giving the business the opportunity to ensure that the Blue Prism solution that is being delivered fully meets their requirements.

The following steps should be taken during UAT:

- The process should be tested against UAT test plans that should have been created by the business based upon the Process Documentation
  - The process should be tested in an environment that mirrors the live environment (i.e. ran on a Virtual Machine (VM), using BP Server and credentials, test mode turned off, scheduling used).
  - Testing should be done in control room. The Operation Handbook should be used to ensure it adequately enables users to run processes.
  - One of the aims of UAT is to ensure that the automated Blue Prism solution will work in the live environment. If possible, UAT tests should be done on live systems using live accounts.
  - It is recommended that UAT testing is performed on a limited number of cases for each scenario, and each case worked by Blue Prism is audited by the tester to ensure it has been worked correctly.
  - 'Stress tests' should be included. For example, system not available or invalid file format type errors.
  - The UAT testers should ensure that exception reporting, performance reporting, and management information reporting all meet requirements
- **Note: If test systems are not available and access to the live systems are not made available for UAT, the initial system exception rate during live roll-out may be high, and additional time will need to be factored in to cater for post-live support and changes.**

## 8. Live Proving

If Blue Prism processes are to be ran in a new or changed environment, testing must be done to ensure the environment is set up correctly. For example, if a new process is moved from a test environment to a live environment.

The following steps should be taken during Environment Testing:

- The environment should be checked to ensure it has been set up in accordance with the Solution Design and Environment Definition Documents.
- All Blue Prism requirements for the process should be checked prior to use (i.e. Work Queues correctly created, Credentials set up, environment variable configuration done, such as setting network paths and test mode etc)
- All desktop or VM settings required are set. For example, access is given to required systems and network paths. Screen lock turned off if required etc.
- Test Rigs (see the Component Testing section) should be ran for each application.
- In a new environment the Blue Prism process should be rolled out gradually. Starting with just running one case at a time and only increasing the number of cases worked per session if the process completes successfully.
- Different environments may run at different speeds. Case times should be evaluated in the new environment to ensure that allocated resources will meet SLAs in the new environment.

## 9. Implementation Testing

When implementing a process into the production environment it should initially be rolled-out with greatly reduced case volumes, with volumes only increased if performance and exception rates are as expected.

Initially, a pre-agreed proportion of cases should be audited to ensure the process is behaving as expected.

As the numbers of cases worked in the production environment is gradually increased, unseen or unexpected scenarios may occur, this may initially cause a higher than anticipated exception rate. The Blue Prism performance report should be used to monitor exception rates and system exceptions with a high occurrence will require fixing in the development environment.

Where adequate system test environments have not been available during development and the previous testing phases, an increased implementation testing period will be required to ensure the process is working as expected in the production environment.

## 10. Regression Testing

Any application, process or environment changes should be fully regression tested. Regression testing should not only ensure that the change made to a process or object works, but also that any unchanged but related objects or processes still work as expected.

The following steps should be taken during Environment Testing:

- If possible, the Blue Prism development team should be included in application change notifications so that adequate time is given to test the changes and make object changes if required. An emergency change methodology should be agreed and used where such notifications are not given or are not available.
- A method of identifying related objects and processes should be in place. This may be done by maintaining a matrix of such relationships. For example, if an action changes within an object it must be possible to identify all other objects and processes that uses that action and therefore needs to be regression tested.
- For any Blue Prism objects or processes that are changed, all the steps outlined in this Test Phases document can be followed for regression testing (for example, test rigs should be used to test changes to objects or components). Depending on the size of the change, less time will be required for regression testing than was required for originally testing the process.
- If a Process is changed the process documentation (and test plans) and solution design should also be updated to keep in line with the actual process. User Acceptance Tests should be re-performed.