IXC

Change-based regression test scope workbook

From a change's impact candidates to the reasons for running or excluding tests

A practical IXC resource for adoption decisions and delivery

Published 2026.09 · English edition · PDF + Excel worksheets

Using this resource

Records the regression tests to select for a single change, the reasons for exclusions and the unverified conditions.

QA leads, engineering leads and product managers who set regression scope

Contents

01 Fix what changed and what the decision covers 3

02 Find impact candidates beyond the direct change 4

03 Record the scope level and reasons to broaden it 5

04 Compare three fictional changes 6

05 Record selection and exclusion for each test 7

06 Pass execution results into the release decision 8

How to complete and reuse it

Read the criteria in this PDF, then fill the blank fields with your actual conditions and anything still unconfirmed. The Excel file for the same practice area has a worksheet for each resource. Keep the worked examples separate from real customer cases and confirmed quotes.

With attribution, you may quote and adapt this material for internal review and your own work. The latest files are at www.ixc.co.kr/en/assets.

Fix what changed and what the decision covers

First, record what changed and why. Do not set execution scope by lines of code alone; check which rules change in user actions, permissions, data and external integrations. If several changes are bundled, connect each change identifier with the boundaries they affect in common.

Change review brief

Item	What to record	Reference and owner
Change identifier and reason		
Requirement versions before and after		
Target build and test version		
User, permission and data impact		
External integration and configuration changes		
Release constraints and reviewer		

> **Scope of the worked examples**
>
> The changes in this resource are fictional cases for practicing scope selection. They are not real customer changes or test execution results. Before applying an example's choices as they are, check your own requirements and the impact on your code and configuration.

This review produces the tests to run and the reasons for exclusions. Selecting a test does not mean it passed, nor is it release approval. Once scope is set, link the actual run identifiers and results to the release decision memo.

Find impact candidates beyond the direct change

Start from the changed screen or function and follow through to shared modules and callers, permission policies, stored data and external consumers. Even when a test is linked to a requirement, review whether the link reflects the current version. Record a judgment of no impact as well, citing the code, documents or owner answers you checked.

Impact candidate review sheet

Impact candidate	Link and change evidence	Failure boundary to check	Review outcome and owner
Worked example: common document permissions	Viewing and downloads that use the same policy	Document access across roles and organizations	Included as a candidate beyond the direct change

Direction for finding candidates	Materials to check
From requirements to tests	Links among changed requirements, cases and defects
From implementation to users	Callers, roles and feature paths
From data to consumers	Persistence rules, lookups and exports
From configuration to shared paths	Environment differences, shared configuration and external integrations

Do not conclude that there is no impact just because no link exists. Leave dependencies you have not yet checked as unresolved, and assign an owner and a time to check them. Traceability review is the step that finds candidates; which of them actually run is decided on the next page.

Record the scope level and reasons to broaden it

Set execution scope by weighing impact reach, change uncertainty, the consequences of failure and controllability together. Even a small permission change broadens the scope if it touches several core flows. Do not let a count of low-risk items average away an important reason to broaden scope.

Scope level	When to consider it	What the execution scope covers
Focused	Impact stays within an independent boundary	The changed flow and confirmed adjacent paths
Expanded	Reaches multiple roles, integrations or data consumers	Affected connections and failure boundaries
Broad / Release-level	Reaches widely into shared policies and core boundaries	The release's important flows, checked together

Scope decision record

Review axis	Evidence checked	Chosen level and reason	Unresolved items and next check
Impact reach			
Change uncertainty			
Consequences of failure			
Detection and rollback			

The scope level does not set the number of tests

Even Broad does not mean running every test without exception. Check important failure boundaries widely, and record the reason for each item excluded as low relevance. If capacity to run tests falls short, do not relabel the scope as lower; record an unverified condition instead.

Compare three fictional changes

The following cases are worked examples of choosing scope by impact boundary rather than by lines changed. Each case has conditions assumed to be already confirmed. In a real project where those conditions have not been confirmed, the same level cannot simply be selected.

Fictional change	Boundary assumed confirmed	Example selection	Exclusions and open checks
A: common document permissions	The same policy applies to viewing, search and downloads	Broad / include role and organization boundaries	Independent statistics screen excluded after confirming no link
B: export field added	Two consumers read the file	Expanded / check the file and both consumers	On hold if the consumer inventory is unresolved
C: independent display text edited	Identifiers, stored values and shared logic unchanged	Focused / check the display and adjacent interactions	Record the evidence that payments and permissions are not linked

> **Example reasoning: even if A's change is a single line**
>
> Because the common policy is called from several places, verification does not end with normal behavior on one screen. Add viewing across role and organization boundaries, search results and downloads to the candidates. Whether each was actually run, and its result, is recorded separately after execution.

If B's consumer inventory is out of date, or C's text also serves as a business code, the assumption breaks. Reopen the scope decision record and review additional impact candidates. Before adopting an example's scope level, first check whether the conditions used for the judgment match your own situation.

Record selection and exclusion for each test

Once a scope level is chosen, work down to actual test identifiers. Record the reasons for selection and exclusion in the same sheet, and distinguish the candidates not yet decided. Keep items whose environment is not ready apart from items excluded as irrelevant.

Execution and exclusion decision sheet

Test or candidate	Selected, excluded or unresolved	Supporting reason	Execution preparation and owner
Worked example A: cross-organization document access	Selected	Direct impact of the common permission policy	Prepare accounts for each organization
Worked example A: independent statistics label	Excluded	Assumes no link to the policy or data	Actual link still to be confirmed

> **Keep a lack of time apart from reasons for exclusion**
>
> If a core permission scenario cannot run because its accounts are missing, do not record it as low relevance. Record the accounts needed, the resolution owner and when it will next be checked, and report the unverified risk to the release owner.

Tests that previously found defects, and real usage flows, also support selection. However, an absence of defect history alone does not mean the risk is low. Check the integrations and data states that are only meaningful when run together, and also review whether complex UI tests can be checked at a more suitable layer.

Pass execution results into the release decision

Once the selection sheet is complete, link the execution environment, the round and where the results are stored. When the build or requirements change, recheck which selection reasons have changed. Separate the owner who records results from the owner who approves release, so that a plan alone is never marked as releasable.

Execution and re-review handover sheet

Selected test and round	Build, environment and run identifier	Result, defects and un-verified items	Next check and owner

Change that requires re-review	Record to revisit
Build, requirement or configuration change	Change brief and impact candidates
New dependency or defect found	Scope level and reasons for selection and exclusion
Delay in account or environment preparation	Unverified conditions and execution schedule
Execution complete or fix verified	Result sheet and release decision memo

PASS, FAIL, BLOCKED and NOT RUN are for actual execution records. READY, READY WITH CONDITIONS and HOLD at the release meeting are a separate decision made after reviewing the agreed criteria, results, unverified scope and residual risk. When discussing scope with IXC, prepare the change list, current cases, unresolved dependencies and release constraints together.

IXC

Publication and use

Based on your change list, current tests, unresolved impact and release constraints, we work with you to set the scope to run and the supporting reasons.

IXC Corp.

CEO Nam Kichul | Business registration no. 109-86-56732
4F 408, Chuncheon ICT Venture Center
7 Huseok-ro 462beon-gil, Chuncheon-si, Gangwon-do, Republic of Korea
+82-33-242-0210 | info@ixc.co.kr

www.ixc.co.kr

www.ixc.co.kr/en/services/qa-process-consulting

www.ixc.co.kr/en/assets/regression-scope-workbook

References

IXC: What should set regression test scope, if not the size of the code change?

https://www.ixc.co.kr/en/insights/risk-based-regression-testing-scope

IXC: Designing traceability so requirement changes stop losing tests

https://www.ixc.co.kr/en/insights/requirements-test-traceability

ISTQB CTFL v4.0.1: regression testing and impact analysis

https://istqb.org/wp-content/uploads/2024/11/ISTQB_CTFL_Syllabus_v4.0.1.pdf

IXC QA Process Consulting

https://www.ixc.co.kr/en/services/qa-process-consulting