PH7X SYSTEMS ®

0.1.0 · CHAPTER 32

Write the test plan before the first test, and schedule no wave until it has passed

SharePoint Migration Field Guide · Chapter 32

pH7x Systems

2026-11-10

32. Write the test plan before the first test, and schedule no wave until it has passed

> A pilot proves that a few sites moved. A test plan proves that the destination works for the sites that were not in it.

Chapter 31 rehearsed the procedure on copies and ran a pilot for the people of a few real sites. Chapter 34 will have each wave accepted by the people who use it. Between the two sits everything that belongs to the destination as a whole and to no single site: its service applications, its search, its sign-in, its integrations, its speed, and the restore the way back depends on. Nobody accepts those in a wave, because by then a failure holds every wave at once. This chapter is where you test them.

> **What has to be tested, in which environment, by whom and against which pass condition, before the first wave is scheduled?**

The decision this chapter leads to is whether the test plan has passed, taken on a written plan in which every test names its environment, its owner and a pass condition written before the run. Until it has, Chapter 33's calendar stays a draft.

Which estates this chapter is for

A SharePoint Server 2016 or 2019 farm moving to SharePoint Server Subscription Edition by database attach, which this chapter treats first and in most depth. A copy into SharePoint Online has its own section near the end, because half of these tests change shape there.

You need four things before you start:

- the rehearsal farm Chapter 23 built and Chapter 31 used, with its trial attach done;
- Chapter 31's pilot record, every finding in it;
- the baseline and the restored backups of Chapter 30;
- the dispositions of Chapters 10, 11, 15, 19, 20, 21 and 22, because each one recorded the test that proves it.

Those chapters own their tests. This chapter gathers them into one plan, adds what none of them covers, and runs the whole against the destination.

What Microsoft documents, and what it leaves to you

We found no Subscription Edition page on planning tests. What Subscription Edition has is two checklists. The page on verifying database upgrades asks whether service applications and their proxies are configured as you want, "Are all features associated with the sites working?", and for search: run queries, "Twenty-four hours later, view the query reports and look for issues", "Search for people and profiles", and check every search customisation (verify database upgrades, checked on 10 November 2026). The page on site collections adds an order and a control: "Begin by validating high-impact or high-profile sites", and "create a new site collection by using a representative set of lists, libraries, Web Parts, and so on" to check that the basic elements work (upgrade a site collection, checked on 10 November 2026).

The method is older. Microsoft's guidance for the upgrade from 2013 to 2016 says: "Run multiple tests", "Do not ignore errors or warnings", and "Test the upgraded environment, not just the upgrade process" (best practices for 2016, checked on 10 November 2026). Its performance testing page, written for SharePoint Server 2013, says what a test plan starts from: "Know the goal of your testing" and "Understand how to measure for your testing goal" (performance testing, checked on 10 November 2026). Neither page lists Subscription Edition among the versions it applies to. We use them for method, never for a version-specific fact, and we do not know of anything in Subscription Edition that contradicts them.

The plan is a table, written before the first run

One line per test. We recommend these columns, and we recommend writing every line before any test runs: a pass condition written after the result takes the shape of the result, as Chapter 34 says of acceptance criteria.

COLUMN	WHAT IT HOLDS
Test	One action a named person can repeat, on named sites or items
Proves	The disposition or requirement it protects, and the chapter that recorded it
Environment	Which farm or tenant, and which copy of the data
Identity	The account that runs it, and why that one
Pass condition	What is observed when it passes, written so that two people would agree
Evidence	What is kept: an output file, a log entry, a screenshot, the other system's log
Owner	Who runs it and who signs the line
Result	Passed, failed with a cause, or not run with a reason and a date

The owner of a line is the person who uses what it protects, or who answers for it: a process owner for a workflow, the owner of the other end for an integration, the database administrator for a restore. The farm administrator signs the farm's own lines and nobody else's.

The test environment, and what it shares with production

Microsoft's 2016 guidance asks for a test environment "as similar as possible to your real environment", with "the same settings, the same URLs, and so on", and for "copies of your actual databases" (best practices for 2016, above). The 2013 performance page puts the cost plainly: "The effort to create a test environment is often underestimated. It should duplicate the production environment as closely as possible" (performance testing, above). For an attach, that environment is Chapter 31's rehearsal farm.

THE TEST FARM	SAME AS PRODUCTION, OR DIFFERENT	WHY
Addresses	Same, reached only from test machines by a name-resolution entry	A different address produces findings the real upgrade will not have
Content	Copies of the real databases, the largest included	Sample data hides the kinds and sizes of sites that fail
Customisations	The set Chapter 15 decided, no more and no fewer	The test is of the farm you will run
Sign-in	The methods Chapter 21 chose, with real accounts in several roles	A test passed by an administrator proves the administrator's access
Service applications	Upgraded from copies, in Chapter 10's order	Their data is what the sites call
Roles and servers	The same roles, and two servers where production has several	Interactions between servers of one role do not appear on one
Server names and SQL Server	Different computers	Directory conflicts, and load on production
External connections, outgoing mail, workflows	Pointed at test systems before anyone signs in	On a copy, the link to the other system is still real

Sources for the table: best practices for 2016 and performance testing, above, and Microsoft's page on trial upgrades to 2013, which asks for the same addresses tested "only from computers that have host file changes", different computer names, "separate servers that run SQL Server", at least two servers per role where production has several, and warns that for external data "the link to the data source is real" (trial upgrade to 2013, checked on 10 November 2026). That page is about the upgrade from 2010 to 2013. Microsoft does not document a test environment for Subscription Edition, so we carry these points forward as our recommendation.

Write every difference between the test farm and production into the plan. A test proves what it ran on, and the list of differences is the list of things it did not prove.

What is tested

AREA	THE TEST, IN ONE LINE	ENVIRONMENT	OWNER	PASS CONDITION
Sites	Each kind of site opens, its lists and libraries read and write, and a new site collection works	Rehearsal farm	Validator of that kind of site	Every task Chapter 31 listed is done as before, or the difference is a finding
Customisations	Each kept solution, web part and script does its job on a real page	Rehearsal farm	Whoever owns it under Chapter 15	Its own specification, exercised by a user who may and one who may not (Chapter 17)
Service applications	Each one answers from a site: a term picked, a profile read, a stored credential used, an external list opened	Rehearsal farm, then the new farm	Owner of each service	Chapter 10's comparison with the old farm, item by item
Search	Chapter 11's tests, after a full crawl of every content source	Rehearsal farm	Search owner	Each dependency's own test; results trimmed to each person's access
Sign-in	Chapter 21's tests: each method, each zone, the People Picker, the crawler	The new farm, before content	Identity owner	A named user per method opens what they should and is refused the rest
Workflows and forms	A new instance of each kept workflow completes; each kept form submits and reads back	Rehearsal farm	Process owner	Chapters 19 and 20's specifications
Integrations	Chapter 22's four results, under the destination identity	Rehearsal farm, against test ends	Owner of the other end	It reads, it writes, it is refused without the grant, and its real job ran
Performance	The baseline's measures, read again on the new farm	A farm with production's hardware	Infrastructure owner	Each measure within the limit its owner wrote beside the baseline
Backup and restore	The backup the way back depends on, restored elsewhere, checked and timed	A SQL Server that is not production's	Database administrator	Restored, clean, and inside the time Chapter 36 needs

The technical verification of a copy is Chapter 8's, and `Test-SPContentDatabase` and the three readings of an attached database are Chapter 23's. They are prerequisites of this table, not lines of it: nothing here starts on a database whose attach was not proved.

The procedure

1. **Freeze the plan.** Every line written, owned and dated; the decider Chapter 1 named has read it. Add each finding of Chapter 31's pilot record as a line of its own, so that what the pilot found on three sites is tested where it could occur on three hundred.
2. **Confirm the environment.** Read the list of differences aloud with the owners. Point external connections at test systems. Take a backup of the test farm's databases as they stand: tests write, and you will restore between runs.
3. **Read the farm.** Record what the destination is before anyone judges what it does (first script below).
4. **Test the sign-in on the new farm before content arrives**, as Chapter 21 sets out, and again on the rehearsal farm with content.
5. **Test the sites and customisations**, the highest-impact sites first, then a new site collection built from the same parts (second script below).
6. **Test each service application from a site that uses it**, then workflows and forms.
7. **Run the full crawl, then test search**, and read the query reports a day later.
8. **Test each integration** against the test end its owner agreed.
9. **Measure performance** against the baseline.
10. **Restore the backup the way back depends on**, check it and time it (third script below).
11. **Record, fix, repeat.** Every failure gets a cause before it gets a fix, and the fix repeats every line it touches.
12. **Decide.**

The scripts below are illustrative. No farm was available to us, so none was run: try each on the rehearsal farm first.

Read the farm before you test it

```
# Illustrative, not run against a farm. Rehearsal farm, SharePoint Subscription Edition Management
# Reads only: service applications, the default proxy group, service instances per server, search
$servers = "<Server1>", "<Server2>"
Write-Host "Reading service applications, proxies, instances and search status into test-farm-*.tx
Get-SPServiceApplication | Out-File .\test-farm-serviceapps.txt
(Get-SPServiceApplicationProxyGroup -Identity "").Proxies | Format-Table DisplayName, TypeName | O
foreach ($s in $servers) { Get-SPServiceInstance -Server $s | Out-File ".\test-farm-instances-$s.t
Get-SPEnterpriseSearchServiceApplication | Get-SPEnterpriseSearchStatus -Text
```

`Get-SPServiceApplication` with no parameter "returns all service applications in the farm"; `Get-SPServiceInstance` with `-Server` returns the instances of one server; the proxy group line is the one Microsoft's upgrade procedure uses to list the default group's proxies with "their display names, type names, and IDs" (upgrade service applications). `Get-SPEnterpriseSearchStatus` gives

each search component one of four states, Active, Degraded, Failed or Unknown, and with `-Text` its output "is printed to the console and cannot be piped to a file", which is why the last line writes no file (all four checked on 10 November 2026).

One limit applies to all three reference pages: they list SharePoint Server 2010 or 2013 to 2019 as applicable, not Subscription Edition, as Chapter 23 found for `Test-SPContentDatabase`. Microsoft's Subscription Edition procedure itself runs `Get-SPServiceApplication` and `Get-SPEnterpriseSearchServiceApplication` in the Subscription Edition Management Shell (upgrade service applications, above). We found no Subscription Edition page that runs the other two, so the rehearsal farm is where you confirm that they answer.

Expected result. Every service application Chapter 10 decided to upgrade or create is in the first file, and none it retired. Every proxy it needs is in the default group. Every search component reads Active. **Validated by** comparing each file, line by line, with Chapter 10's record of the old farm. A component that reads Degraded is, in Microsoft's words, typically "a transient situation related to a restart or network issues": read it again before you record it.

Test the sites, the highest-impact first

Subscription Edition upgrades site collections with their database by default and has no "site collection compatibility modes", so a site is on the new version or it is not (upgrade a site collection, above). The same page gives the PowerShell that reads one site collection's upgrade session:

```
# Illustrative, not run against a farm. Rehearsal farm; reads each site collection's upgrade sessi
$sites = "https://<address>/sites/<site1>", "https://<address>/sites/<site2>"   # the plan's test
foreach ($url in $sites) {
    Write-Host "Reading the upgrade session of $url; copy the output into the plan's evidence"
    Get-SPSiteUpgradeSessionInfo -Site $url
}
```

Then the people test. Each validator does the tasks Chapter 31 listed for that kind of site, on items created for the test, and records done as before, done differently or not done. Microsoft's checklists for upgraded sites give the things to look at on every page: images, style sheets, themes, JavaScript controls and script errors, and any large list, because "large list query throttling is turned on" by default (upgrade a site collection, above).

Expected result. Each session reports the site collection upgraded without errors, every listed task is done as before, and a new site collection built from the same lists, libraries and web parts works. **Validated by** the validator's signed line, with the session output beside it. A clean session and a failed task is a failed line.

Test search after the crawl, not before

The new farm's index "is empty" until "a full crawl of the entire indexed corpus" has run (upgrade service applications, above), so every search test before it tests nothing. On the Manage Content Sources page a content source's status reads "Crawling Full" while it runs (start, pause, resume, or

stop a crawl, a page that lists Subscription Edition), and Microsoft's walkthrough waits "until the value in the Status column is Idle. That means that the full crawl has finished" (enable crawling of your content, both checked on 10 November 2026).

Expected result. Each of Chapter 11's dependencies passes its own test, a named person finds what they may open and nothing else, and people search returns real names. **Validated by** the same queries run by the same people on the old farm, as Chapter 31 compared them, and by the query reports a day later. Time the full crawl: Chapter 33 needs the figure.

Measure performance on the baseline's own measures

Chapter 30 took the baseline on the old farm. Read the same measures the same way on the new one, over the same kind of window, and compare measure by measure. Microsoft's 2013 page names the pair for throughput: "you'll want to measure the RPS and page latency" (performance testing, above). Three of its points hold whatever tool you use:

- **Test custom code alone first.** "Custom components are often a major culprit of performance and reliability problems in production systems", so the page tests them "in isolation first", then the farm with and without them.
- **A page that loads has not passed.** An error page can answer with "an HTTP status code of 200 (successful)", so a timed request checks what came back, not only that something did.
- **Use several roles.** Behaviours "such as output caching work differently depending on the rights of the current user", so a test run only by an administrator measures a path most people never take.

That page describes load tests built with Visual Studio Team System, and our search of Microsoft Learn found no later page for 2016, 2019 or Subscription Edition. We would not build a load test for an attach onto hardware at least equal to the old farm's. We would time the validators' own tasks on both farms, read the baseline's measures on the rehearsal farm while they work, and leave real load as an open line for hypercare. Where the new farm's design differs from the old one, a load test is the only thing that answers, and its tool is your choice: we do not know of a current Microsoft recommendation.

Expected result. Each measure is within the limit written beside the baseline before the test. **Validated by** the infrastructure owner, on the two sets of figures side by side. A rehearsal farm on smaller or virtual hardware cannot pass or fail this line: Microsoft's own page says virtual environments "might not have the same performance metrics as physical environments" (trial upgrade to 2013, above). Record the line as not established, with the date the production farm's hardware can answer it.

Restore the backup the way back depends on

On an attach, the way back is the old farm. Microsoft's procedure says the backups taken before the upgrade guarantee "that you have the data in a safe state if you must enable the original farm again", and asks you to "keep a copy of your original backups in reserve" (copy databases, checked on 10

November 2026). Chapter 30 proved that a backup restores. This test asks a narrower question for Chapter 36: can the databases the old farm needs be restored, checked and back in service inside the time a rollback is allowed?

```
# Illustrative, not run against a server. A SQL Server instance that is NOT production's; needs th
$sql = "<TestSqlInstance>"
$bak = "<\\server\share\ContentDatabase.bak>"
$db  = "<RestoredDatabaseName>"
Write-Host "Verifying $bak on $sql; this reads the backup and restores nothing"
Invoke-Sqlcmd -ServerInstance $sql -Query "RESTORE VERIFYONLY FROM DISK = '$bak';" -QueryTimeout <
# Now restore $bak as $db with SQL Server's own tools, as Microsoft's copy procedure does, noting
Write-Host "Checking the restored copy $db on $sql; this reads the database and repairs nothing"
$t = Measure-Command { Invoke-Sqlcmd -ServerInstance $sql -Database $db -Query "DBCC CHECKDB WITH
Write-Host "DBCC CHECKDB on $db took $($t.TotalMinutes) minutes"
```

`RESTORE VERIFYONLY` checks "that the backup set is complete and the entire backup is readable" and "does not attempt to verify the structure of the data" (RESTORE VERIFYONLY). `DBCC CHECKDB WITH NO_INFOMSGS` checks the current database and, when it finds nothing, returns "The command(s) completed successfully." (DBCC CHECKDB). `Invoke-Sqlcmd` runs both and shows SQL Server's messages with `-Verbose`; `Measure-Command` times the second (all four checked on 10 November 2026). Set the timeout to cover your largest database.

Expected result. The verify succeeds, the restore completes, the check returns no error, and the three times together fit the time Chapter 36 allows. **Validated by** the database administrator, with the output and the times kept. Microsoft's backup guidance for SharePoint Server, a page that lists Subscription Edition, gives the reason for timing it: a trial restoration "can expose hardware problems that do not come up with software verifications" and shows whether "the recovery time objectives (RTO) are met" (backup best practices, checked on 10 November 2026).

A clean restore proves the database, not the farm around it. Where you keep a non-production farm of the old version, we would attach the restored copy there and open its sites. Where you do not, write that the way back was proved to SQL Server's level only, and let Chapter 36 decide what that is worth.

A copy into SharePoint Online

The plan and its columns stay; four areas change.

- **Environment.** The destination is the tenant that will receive the waves, in sites nobody uses yet, as Chapters 30 and 31 set out. There is no farm to read and no service application to compare: the first script has no equivalent here.
- **Sites and customisations.** A copy translates. Workflows and forms arrive rebuilt or not at all, so each is tested against its specification (Chapters 19 and 20), and identities against their mapping (Chapter 24).
- **Performance.** Microsoft's page is direct: "load-testing isn't permitted on SharePoint in Microsoft 365", "we automatically throttle any load tests", and one that runs could give "disappointing and

potentially misleading results" (capacity planning and load testing, checked on 10 November 2026). The test is the validators' own tasks, timed from their own desks on both sides, as Chapter 34 does.

- **Backup and restore.** The way back is the read-only source farm (Chapter 9). The third script applies unchanged, to the source farm's backups, for as long as that farm is the fallback.

Search has no crawl to start: Chapter 31 records when known items first return. Integrations follow Chapter 22 against the tenant, at real volume.

Common failures, and where to look

WHAT YOU SEE	WHERE TO LOOK FIRST	SOURCE
A page does not render	Open `/_layouts/settings.aspx` on the site. If it works, look at the master page or home page; if not, read the site collection upgrade log	Upgrade a site collection
A view returns nothing on a large list	List view throttling, on by default: an index, a filtered view or folders	Upgrade a site collection
Images, styles or scripts are missing	The links to them, then the master page	Upgrade a site collection
Search returns nothing	Whether the full crawl has finished, then each component's state	Upgrade service applications; `Get-SPEnterpriseSearchStatus`
A site cannot reach a service that exists	Whether its proxy is in the default group (Chapter 10)	Upgrade service applications
Stored credentials fail	The Secure Store pass phrase restored from the old farm (Chapter 10)	Upgrade service applications
A feature the old farm ran is off	The switches Microsoft's 2026 updates turned off (Chapter 34 lists them)	Chapter 34
A test passes for you and fails for its user	The identity that ran it; repeat as a user with ordinary rights	Performance testing
The verify passes and the check fails	Expected: the verify does not read the data's structure. The backup is not usable as it stands	`RESTORE VERIFYONLY`

Pages as cited above, all checked on 10 November 2026. When a test fails only on the test farm, look first at the list of differences from production: Microsoft's 2016 guidance warns that each difference means "time resolving unrelated issues" (best practices for 2016, above).

Risks and limits

- **A test farm with real links writes to real systems.** Check where every connection, workflow and mail setting points before the first person signs in, as Chapter 31 does, and agree every run into another system with its owner first.
- **Tests change the data they test.** Microsoft's performance page restores the content from a backup after each load test so that results stay comparable (performance testing, above). We would do the same between rounds of this plan.
- **A command aimed at production costs production.** The trial upgrade page warns that running `Test-SPContentDatabase` "against a live database, instead of a copy" may affect production's performance (trial upgrade to 2013, above). Every script here runs on copies or on a server that is not production's.
- **The guidance is older than the product.** The method pages are written for 2013 and 2016, and the cmdlet references stop at 2019. Where a result on your rehearsal farm disagrees with a page, the farm is right about your estate, and the difference is a finding.
- **Some things no test farm shows.** Real load, a full month of use, the query reports of real traffic. Leave each as an open line with an owner and a date in hypercare, and say so in the decision.

Passed, passed with findings, or not passed

The decider reads the plan line by line with the owners, never a count of green lines, and signs one of three:

- **Passed.** Every line passed, and Chapter 33's calendar may be signed.
- **Passed with findings.** Each failed line is fixed and repeated with everything the fix touches, or accepted in writing by the owner of what it affects. Each line that only production can answer is open, with an owner and a date before the wave or the period that needs it.
- **Not passed.** A line no fix meets returns to the chapter that recorded its disposition, and no wave is scheduled.

A line that passed on a second run with no explanation of the first has not passed. Find the cause, or record it as a finding.

Then the plan hands on what the next chapters need:

- **Chapter 33** receives the measured durations a wave must hold: the full crawl, the day before the query reports, the validators' hours per kind of site.
- **Chapter 34** receives the tests themselves. Each wave's acceptance record draws its lines from this plan, so that a validator meets a site's defects there, never the farm's.
- Chapter 35 receives the signed plan with its open lines, as one of the criteria the cutover decision reads.
- **Chapter 36** receives the restore's times, its evidence, and how far the way back was proved.

- Chapter 37 receives every accepted finding and every open line, with the baseline's measures to read again under real load.

A wave is the wrong place to find out that the farm does not work. Test the destination once, for everyone, before anyone's site depends on it.

Out of scope

The trial attach, the rehearsal and the pilot are Chapters 23 and 31. Verifying what a copy moved is Chapter 8. Each wave's acceptance is Chapter 34, the cutover Chapter 35, the rollback procedure Chapter 36 and hypercare Chapter 37. The tests of each disposition stay with the chapter that recorded it: this chapter runs them, it does not rewrite them. Automated interface testing, load-test tooling, security and penetration testing, backup and recovery design, third-party tools and licensing are not covered.

Before you continue

- [ ] Every line of the plan has a test, an environment, an identity, an owner and a pass condition, all written before the first test ran
- [ ] The test farm shares production's addresses, sign-in, customisations and real data, and every difference from production is listed in the plan
- [ ] Sites, customisations, service applications, workflows and forms were tested by the people who use them, never only by the farm administrator
- [ ] Search was tested after the full crawl finished, and each integration under its destination identity against an end its owner agreed
- [ ] Performance was compared with the baseline measure by measure, and every line only real load can answer is open with an owner and a date
- [ ] The backup the way back depends on was verified, restored on another server, checked and timed against what Chapter 36 needs
- [ ] Every failure has a cause, a fix and a repeat of every line the fix touches, and no line passed on an unexplained rerun
- [ ] The decision is signed on the lines, and Chapters 33 to 37 hold what this plan hands them