Lecture 2. Project Planning and Control

1. Project Scope, Requirements and Work Breakdown Structure
2. Project Scheduling and Time Estimation
3. Resource Planning and Project Budgeting
4. Project Risk and Quality Planning
5. Monitoring Project Progress and Managing Changes

1. Project Scope, Requirements and Work Breakdown Structure

Project planning turns a general project idea into an agreed description of what the team will deliver and how the necessary work will be organised. Before estimating time or costs, the team needs to understand the expected results, the requirements these results must meet, and the boundaries of its responsibility.

In language-related projects, an apparently simple request such as "localise the admissions portal" can involve very *different expectations*. The client may expect translated webpages, adapted interface messages, a terminology database, testing and support after launch. The project team therefore needs to make these expectations explicit.

Running example: Localising a university admissions portal

Assume that the university wants Ukrainian and Polish versions of an existing English-language portal. The agreed content includes 20 webpages and 40 interface messages. The existing content management system supports multiple languages.

These quantities and technical conditions are illustrative assumptions. They allow us to practise defining and organising project work.

Understanding project scope

Two related concepts help us distinguish the characteristics of the intended result from the work required to produce it.

Product scope describes the features and characteristics of the product, service or result. In our example, it includes the agreed language versions, the content available in each language, and the linguistic and technical characteristics of the localised portal.

Project scope covers the work needed to deliver that result. It includes clarifying requirements, preparing terminology resources, translating and reviewing content, integrating it into the portal, testing the localised versions and handing them over to the university. Project coordination and documentation also require work and must be included.

A ***deliverable*** is a specific output that can be checked and accepted. Examples include an approved glossary, a reviewed Polish content package or a localisation test report.

An ***activity*** is an action performed to produce a deliverable: extracting terminology, translating interface messages or checking hyperlinks.

This distinction makes planning more precise. "Translate the admissions pages" describes an activity. "Reviewed Polish versions of the 20 agreed admissions pages" identifies a deliverable and makes its expected coverage clearer.

Identifying and documenting requirements

A ***requirement*** states a condition or capability that the project or its result must satisfy. Requirements translate stakeholder needs into statements that can guide the work and support verification.

For example, an applicant's need may be: "I want to understand which documents I must submit." A corresponding requirement could be: "The Ukrainian and Polish versions must include the complete document checklist from the approved source pages."

Product requirements commonly include:

- ***Functional requirements****:* what the product must enable users to do, such as switch between language versions.
- ***Non-functional requirements****:* qualities the product must demonstrate, such as readability, terminology consistency or correct display of characters.

The team should also record project requirements and constraints, such as the required handover format, an agreed launch date or the obligation to use the university's existing platform.

Requirements are developed through ***elicitation****:* actively discovering and clarifying what stakeholders need. For a language project, useful methods include reviewing the project brief and source materials, interviewing the client, observing users, holding a clarification workshop and discussing sample translations or interface mock-ups. The choice of method should match the uncertainty that needs to be resolved. A short review of a sample page may reveal expectations that a general questionnaire misses.

In the portal project, the admissions office can clarify the meaning of admissions terminology, applicants can identify confusing instructions, and the technical team can explain platform limitations. Their contributions address different aspects of the requirements.

A useful requirement should be clear, feasible, relevant to the project objective and verifiable. Statements such as "the translation must be excellent" need clarification: what will reviewers examine, and what evidence will demonstrate that the requirement has been met?

Requirements and acceptance criteria

Acceptance criteria specify the conditions that a deliverable must meet before the authorised client representative accepts it. They should be agreed early enough to guide production and review.

The following examples show how broad expectations can become checkable conditions.

Requirement	Example acceptance criterion
The agreed content must be available in Ukrainian and Polish.	Each language version contains all 20 webpages and 40 interface messages listed in the approved content inventory.
Admissions terminology must be consistent.	Designated terms match the approved glossary; any exceptions are documented and approved by the terminology reviewer.
The translation must preserve essential admissions information.	The final review identifies no unresolved errors that alter eligibility conditions, deadlines or required documents.
Localised text must display correctly.	No missing characters, clipped text or overlapping labels remain in the agreed browser and screen-size tests.

The team must also identify who checks each criterion and who has authority to accept the deliverable. A translator may confirm that translation is complete, while the admissions representative confirms that the information is accurate and suitable for publication.

These criteria establish what will be checked. The detailed organisation of quality reviews and testing will be considered later in the topic.

Agreeing the project boundaries

The ***project scope statement*** records what the project will deliver and the boundaries within which the team will work. It provides *a shared reference* for the client and the project team.

For our example, a concise scope statement could read:

The project will deliver Ukrainian and Polish versions of the 20 webpages and 40 interface messages identified in the approved content inventory. It includes requirements clarification, terminology and style resources, translation, linguistic review, integration into the existing portal, localisation testing, publication and handover documentation. The university will approve the admissions information and nominate representatives to review and accept the deliverables.

This description should be accompanied by explicit exclusions, assumptions and constraints.

Exclusions identify work outside the agreed scope. In this example, they could include additional languages, translation of unlisted downloadable documents, redesign of the admissions process, development of new application functions and ongoing content maintenance after handover.

Assumptions are conditions treated as true for planning purposes but still requiring confirmation. For example, the team may assume that the source content will be approved before translation begins and that reviewers will be available when needed.

Constraints are limits that affect the team's choices. Examples include a fixed launch date and the requirement to use the existing content management system.

An assumption can prove incorrect. If the source content is repeatedly revised during translation, the team may need to adjust its estimates and agree how to handle the additional work. *Recording assumptions makes these dependencies visible.*

Creating a Work Breakdown Structure

Once the deliverables and boundaries are sufficiently clear, the team can develop a ***Work Breakdown Structure (WBS)***.

A WBS organises the agreed project work into a hierarchy of progressively smaller components. This process is called ***decomposition***. The lowest-level components are ***work packages***: portions of work that can be assigned, estimated and monitored meaningfully. *There is no universal number of levels; the necessary detail depends on the project.*

To develop a WBS, the team should:

1. Identify the major deliverables and supporting project work.
2. Break each major component into smaller, clearly defined components.
3. Continue until the work packages are manageable.
4. Check that all agreed work is covered and that responsibilities do not overlap.
5. Review the structure with the people who will perform and accept the work.

For the portal project, the following table represents a simplified hierarchy. The whole project is ***1.0 Admissions portal localisation***; the first column contains its major components, and the second contains their work packages.

Major WBS component	Work packages
1.1. Project management and requirements	**1.1.1** Approved scope and requirements record; **1.1.2** Project plans and control records
1.2. Language resources	**1.2.1** Approved English–Ukrainian–Polish glossary; **1.2.2** Localisation style guide
1.3. Localised content	**1.3.1** Reviewed Ukrainian content package; **1.3.2** Reviewed Polish content package
1.4. Integrated and tested portal	**1.4.1** Configured and integrated language versions; **1.4.2** Localisation testing, corrections and test evidence
1.5. Release and handover	**1.5.1** Published and accepted portal release; **1.5.2** Handover documentation and completion record

The codes identify positions within the hierarchy. For example, ***1.3.2*** belongs to ***1.3. Localised content***. *They do not indicate the order in which work must be performed.*

The structure includes supporting outputs as well as the final portal. Terminology preparation, project coordination, testing and handover all require resources, even when the client's initial request mentions only translation.

Checking the quality of a WBS

The *100% rule* means that the WBS must cover all work within the agreed scope. At each level, the lower-level components must collectively cover their parent component. The structure should also avoid including work outside the agreed boundaries. Project management work belongs within this coverage.

For our example, a WBS containing only Ukrainian and Polish translation packages would be incomplete: it would omit terminology resources, integration, testing and handover.

Work packages should also have clear boundaries. For instance, linguistic review of translated files belongs to the localised content packages, while checking how those translations appear and function inside the portal belongs to localisation testing. The descriptions should specify where corrections are handled so that the same work is not counted twice.

Decomposition should stop when further detail offers little management value. Creating a separate work package for every sentence would usually make this project harder to manage.

Describing work packages in a WBS dictionary

A short label cannot explain everything about a work package. A *WBS dictionary* provides the additional detail needed to understand its content, responsibility and completion conditions.

For example:

- *Code and name:* 1.3.2 Reviewed Polish content package.
- *Included work:* translating and linguistically reviewing the agreed 20 webpages and 40 interface messages, including corrections arising from that review.
- *Responsible owner:* Polish language lead.
- *Inputs:* approved source content, glossary and localisation style guide.
- *Completion conditions:* the package is complete against the inventory, follows the agreed terminology and has no unresolved errors affecting essential admissions information.
- *Boundary*: technical integration and testing inside the portal are covered under component 1.4.

The owner coordinates completion of the package; several people may contribute to its delivery.

Connecting requirements, the WBS and subsequent planning

The team should be able to trace each agreed requirement to the deliverable that addresses it and the evidence used to check it. This connection can be recorded in a simple *requirements traceability matrix*, alongside the requirement's source and status. It helps the team check that requirements have been covered and understand which outputs may be affected by a change.

For example, the requirement for consistent terminology links to the approved glossary, the reviewed language packages and the linguistic review records.

In a predictive planning approach, the approved scope statement, WBS and WBS dictionary together form the *scope baseline*. This provides a reference for assessing progress and proposed changes. Approval does not mean that the scope can never change; it means that changes need to be evaluated and authorised.

The WBS then supports scheduling. For the Polish content package, the team can identify activities such as translation, editing, correction and final review. It can estimate their durations, determine dependencies and assign dates. The WBS defines the work to be covered; the schedule adds sequence and timing.

In adaptive projects, detailed scope develops progressively. The same need for clear boundaries, understandable requirements and checkable completion conditions remains, while detailed planning focuses on the work approaching delivery.

Practical task

A university department wants to create an English–Ukrainian terminology database for one academic subject. Working in small groups, define the intended users and main deliverables, formulate three requirements with acceptance criteria, and identify two exclusions. Then create a simple WBS and describe one work package. State any assumptions you make and explain how you would decide that the package is complete.

2. Project Scheduling and Time Estimation

A project schedule connects the work identified in the WBS to time. It shows when activities can start and finish, which activities depend on others, and when important project results should be available.

For our admissions portal project, the team has agreed to deliver Ukrainian and Polish versions of 20 webpages and 40 interface messages. Scheduling now requires decisions about the order of the work, its likely duration and the availability of the people involved.

A realistic schedule develops from this information. It also provides a reference for tracking progress and assessing the consequences of delays.

From work packages to activities and milestones

Scheduling begins by identifying the ***activities*** needed to complete each work package.

For example, the work package *Reviewed Polish content package* includes translating the source content, reviewing the translation and correcting identified errors. These activities may involve different people and have different dependencies.

Each activity should have a clear description, an estimated duration and an identifiable completion condition. “Translate the agreed Polish interface messages” is more useful for scheduling than “Work on localisation.”

A ***milestone*** marks a significant event or achievement, such as:

- Source content approved.
- Both language packages accepted.
- Localised portal released.

Milestones have ***zero duration*** because they represent points in time. The work required to reach them must be scheduled separately. Reviewing a translation takes time; recording that the translation has been approved marks a milestone.

Distinguishing effort from duration

An accurate estimate requires a distinction between the amount of work and the time needed to complete it.

Effort is the total work required, usually expressed in person-hours or person-days.

Duration is the number of working-time units over which an activity takes place, according to the relevant work calendar.

Suppose reviewing a translation requires 16 person-hours. If one editor can spend four hours per working day on the project, the estimated duration is:

$$\text{Duration} = \text{Effort} / \text{Daily availability} = 4 \text{ working days}$$

This simplified calculation assumes a stable work rate and no additional waiting time. The editor's availability matters: being employed full-time does not mean being available to this project for eight hours each day.

Elapsed time includes non-working periods between the activity's start and finish. An activity requiring two working days may span a weekend. Teams therefore need to specify whether an estimate refers to working days or calendar days.

Adding people does not automatically reduce duration proportionally. Some work can be divided, while other work requires a consistent editorial approach, coordination or sequential review.

Identifying dependencies

A ***dependency*** describes a relationship that limits when an activity can start or finish.

The activity on which another activity depends is its ***predecessor***. The dependent activity is its ***successor***.

Four standard relationships can be used.

Dependency	Meaning	Example
Finish-to-Start (FS)	The successor cannot start until the predecessor finishes.	Translation starts after the source content has been approved.
Start-to-Start (SS)	The successor cannot start until the predecessor starts.	Review of released translation batches begins after translation has started and the first batch is available.
Finish-to-Finish (FF)	The successor cannot finish until the predecessor finishes.	Review of all translated content cannot finish before translation of that content is complete.
Start-to-Finish (SF)	The successor cannot finish until the predecessor starts.	Temporary support coverage cannot end before the replacement support service starts.

Start-to-Start and Finish-to-Finish relationships establish limits; they do not necessarily require simultaneous starts or finishes. Start-to-Finish is less common and is mainly useful in situations involving a handover or replacement service.

Some dependencies arise from the nature of the work. Others reflect the team's chosen approach. For example, translating only after the entire glossary is approved may be a planning choice; another project might approve terminology in batches. The team should understand the reason for each dependency before deciding whether it can be changed.

Lead and lag

A ***lag*** introduces a waiting interval between linked activities. A two-day lag might represent an agreed waiting period before a client review begins.

A ***lead*** allows linked activities to overlap. For example, editing may begin before translation of the whole package is complete. Scheduling tools can represent this through a Finish-to-Start relationship with a lead.

For language projects, dividing content into identifiable batches often makes this overlap easier to manage. The reviewer can work on an approved batch while the translator continues with the next one. The schedule should specify what must be available before review starts.

Where a waiting period contains actual work, such as reviewing and approving terminology, it is usually clearer to represent that work as an activity with an owner and duration.

Estimating activity durations

Duration estimates should reflect the amount and complexity of work, the required quality, available resources and relevant previous experience.

Several methods can support estimation:

- *Expert judgement:* asking people familiar with the work to estimate its duration and explain their assumptions.
- *Analogous estimating*: using a comparable previous project, with adjustments for differences in scope or complexity.
- *Parametric estimating*: applying a work rate to a measurable quantity, such as words, interface messages or corpus records.
- *Bottom-up estimating*: estimating detailed activities and using them to construct the schedule.
- *Three-point estimating*: considering optimistic, most likely and pessimistic scenarios.

These methods can be combined. An estimate based on word count can be checked by an experienced translator and compared with records from similar assignments.

For an illustrative parametric estimate, assume that a translation package contains 8,000 words and the assigned translator can complete 2,000 words per project working day:

Estimated translation duration = 8,000 /2,000 =4 working days

The assumed rate is an example, not an industry benchmark. Terminology research, editing and testing must be included separately unless the rate explicitly covers them.

When combining activity estimates, the team must account for dependencies and parallel work. Adding every activity's duration together does not necessarily produce the project duration.

Using PERT to consider uncertainty

The ***Program Evaluation and Review Technique (PERT)*** uses three duration estimates:

•Optimistic estimate (o): a favourable but plausible scenario.

•Most likely estimate (m): the most plausible duration under expected conditions.

•Pessimistic estimate (p): an unfavourable but plausible scenario.

The conventional PERT weighted estimate is:

$$t_e = (o+4m+p) /6$$

For the Polish translation activity, suppose the team estimates three working days under favourable conditions, four as the most likely duration and eleven if substantial terminology difficulties arise:

$$t_e = (3+(4x4)+11) / 6 = 5 \text{ working days}$$

The longer pessimistic scenario raises the weighted estimate above the most likely value. PERT makes this uncertainty explicit, but the resulting estimate is not a guaranteed completion time. A probability of completing the entire project by a particular date requires further analysis of the network and its uncertainties.

Running example: Building the activity network

The following simplified schedule covers the delivery of the localised admissions portal. Project coordination continues throughout and is omitted from this delivery network for clarity.

Assume that:

•All durations are illustrative estimates in working days.

•The two language streams have separate translators and reviewers.

•All dependencies in this example are Finish-to-Start, with no lead or lag.

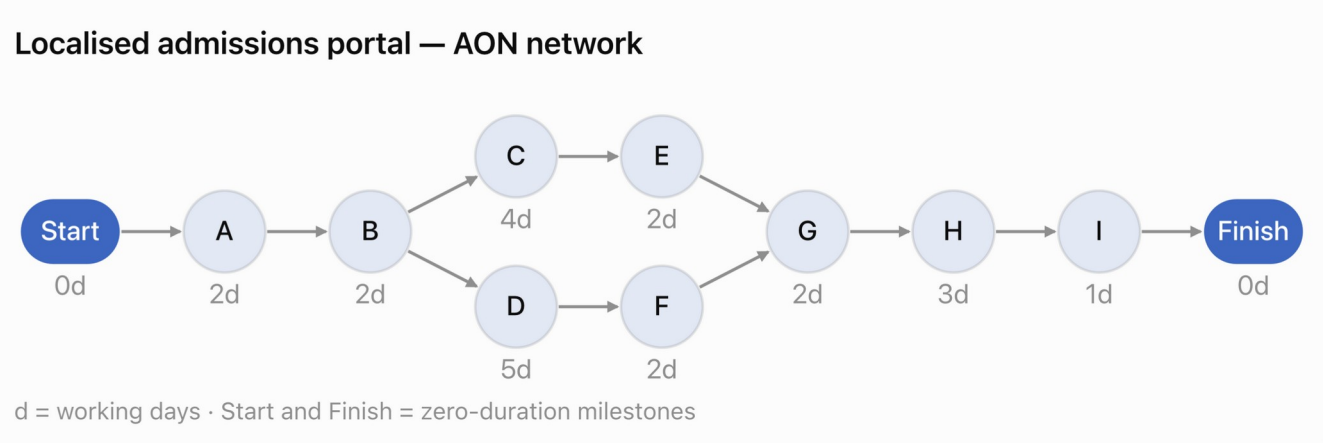
•Integration begins after both language packages have been reviewed and corrected.

•The required people are available when their activities are scheduled.

ID	Activity	Duration	Predecessor
A	Confirm source content and inventory	2 days	None
B	Prepare and approve glossary and style guide	2 days	A
C	Translate Ukrainian content	4 days	B
D	Translate Polish content	5 days	B
E	Review and correct Ukrainian content	2 days	C
F	Review and correct Polish content	2 days	D
G	Integrate both language versions	2 days	E and F
H	Test the portal and correct defects	3 days	G

I	Complete approval, release and handover	1 day	H

An ***activity network diagram*** makes these relationships visible. In an Activity-on-Node diagram, nodes represent activities and arrows represent dependencies.



The diagram shows that the language streams can run in parallel. Integration must wait until both are complete.

Understanding the critical path

The ***Critical Path Method (CPM)*** identifies the sequence of linked activities that determines the earliest project completion under the schedule's assumptions. In this basic network, it is the path with t*he longest total duration.* Activities on that path have no scheduling flexibility without affecting completion.

There are two complete paths in our example.

The Ukrainian path: A-B-C-E-G-H-I
2+2+4+2+2+3+1=16 working days

The Polish path: A-B-D-F-G-H-I
2+2+5+2+2+3+1=17 working days

The *critical path is A–B–D–F–G–H–I*, and the calculated project duration is *17 working days*.

This result has practical implications. If Polish translation takes one additional day and nothing else changes, the project finishes one day later. Completing Ukrainian translation earlier would not, by itself, bring the release forward because integration would still be waiting for the Polish stream.

"Critical" refers here to an activity's effect on the completion date. Activities outside the critical path can still be essential to project quality and acceptance.

Float and scheduling flexibility

Total float, also called ***slack***, is the amount of delay an activity can absorb without delaying project completion or violating a relevant schedule constraint. In this example, activities C and E each have one day of total float.

The Ukrainian stream can therefore finish one working day later while integration still starts as planned. However, this is one shared day of flexibility along that stream. The team cannot add one day to translation and another day to review without affecting completion.

Free float concerns immediate successors: it is the delay possible without postponing the earliest start of a following activity.

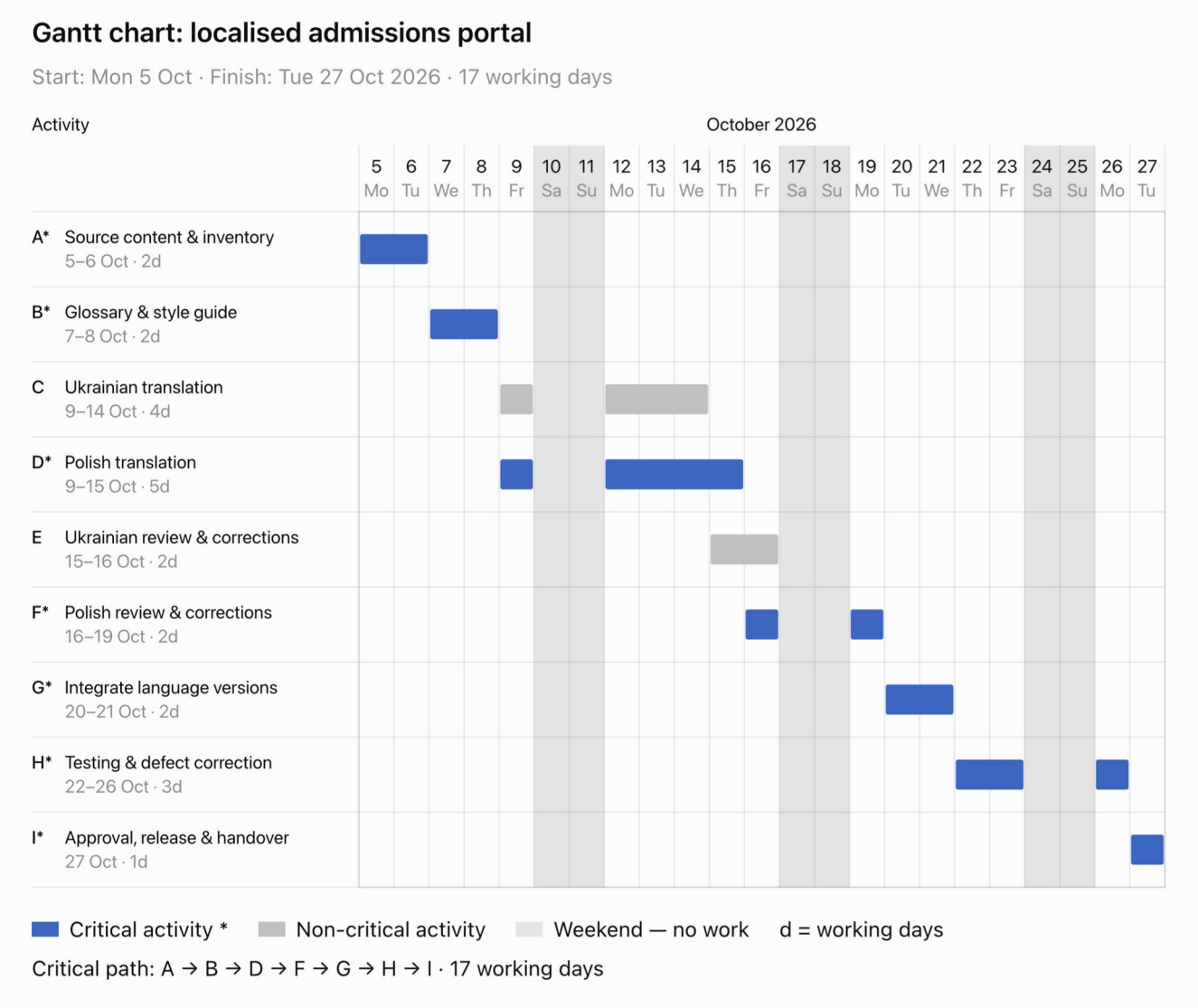
In our example, Ukrainian translation has zero free float because a delay postpones Ukrainian review. The review activity has one day of free float because integration is already waiting for the Polish stream.

Float is calculated from the network. It differs from a deliberately planned time reserve for uncertainty. As estimates and actual progress change, float and the critical path should be recalculated.

Presenting the schedule in a Gantt chart

A ***Gantt chart*** displays activities as horizontal bars on a time axis. Their positions show scheduled starts and finishes, while overlapping bars reveal parallel work. Dependencies, milestones and progress can also be displayed.

To place our example on a calendar, assume an illustrative start date of *Monday, 5 October 2026*, with weekends excluded and no other non-working days. The calculated finish is the end of *Tuesday, 27 October 2026*.



Critical activities are highlighted. Some bars span weekends, but those days are excluded from their working durations.

The chart helps readers see when work occurs. Its reliability depends on the activity estimates, dependencies and calendars behind it. Entering dates without checking these relationships can produce a schedule that looks organised but cannot be achieved.

Checking the schedule before approval

Before agreeing the schedule, the team should check whether the required resources are available, client reviews have been allowed for, and all necessary corrections and handover work are covered.

In our example, parallel language work is possible because separate resources are assumed. If one reviewer must review both languages, the schedule needs to reflect that limitation. Resource availability can change the completion date. Scheduling software calculates dates using dependencies, calendars, constraints and other scheduling information; those inputs must represent the actual project conditions.

Once approved, the *schedule baseline* provides a reference for comparing planned and actual progress. Updated forecasts should show the current expected completion date; changes to the approved baseline require the agreed change process.

Practical task

Using the portal schedule, assume that Ukrainian translation takes six working days instead of four, while all other estimates remain unchanged. Identify the new critical path, calculate the revised project duration and update the Gantt chart. Explain one feasible action the project manager could consider to protect the original completion date.

3. Resource Planning and Project Budgeting

A project schedule becomes achievable when the necessary people, tools and funding are available at the right time. Resource planning identifies what the project needs, while budgeting estimates the cost of providing those resources.

For our admissions portal project, the team has defined the scope and developed a 17-working-day schedule. The next step is to check whether the proposed activities can be supported by actual resource availability and an affordable budget.

Identifying resource requirements

Project resources include more than the people performing the main technical work. A language-related project may require:

- *People and expertise:* translators, editors, terminology specialists, technical staff, testers and a project manager.
- *Tools and infrastructure:* computer-assisted translation tools, terminology databases, collaboration software and a testing environment.
- *Information and access:* approved source materials, reference documents, platform access and the availability of client representatives for clarification and approval.

The WBS and activity list provide the starting point. For each activity, the team should identify the required competencies, the amount of work, the tools or services needed and the period during which these resources must be available.

For example, “one reviewer” is an incomplete resource specification. The reviewer must have the relevant language competence, understand the subject matter and be available during the scheduled review period.

A simple resource plan should record:

1. The activity or work package requiring the resource.
2. The required role, competence or resource type.
3. The estimated quantity, such as person-hours or software licences.
4. The dates and amount of availability.
5. The cost basis and whether the resource commitment has been confirmed.

A *resource calendar* records when a particular resource is available. It can reflect working days, leave, other assignments and limits on daily participation.

Matching workload to available capacity

Resource planning requires checking both the total amount of work and its distribution over time.

Suppose Ukrainian translation requires 24 person-hours and the translator can contribute six hours per working day:

Required duration = 24 / 6 =4 working days

This supports the four-day estimate used in our schedule. If the translator can contribute only three hours per day, the same workload would require eight working days under otherwise unchanged conditions.

Availability must therefore be expressed clearly. A person may be assigned to the project while also teaching, supporting another client or participating in other organisational activities.

An *overallocation* occurs when the work assigned to a resource exceeds its available capacity during a particular period. A person can have enough hours available across the whole project and still be overallocated on a specific day.

Running example: A shared reviewer

The original portal schedule assumes separate reviewers for the two language streams:

- Ukrainian review and correction, activity E, takes place on working days 9–10.
- Polish review and correction, activity F, takes place on working days 10–11.

Assume that each activity requires eight person-hours, distributed as four hours per day.

Now suppose the team proposes using one person who is qualified to review both languages but is available for only four hours per day. On day 10, the combined demand would be:

4 hours for Ukrainian + 4 hours for Polish =8 hours

The reviewer has only four available hours, so the original allocation is infeasible.

If Ukrainian review is completed first, Polish review moves to days 11–12. Integration then moves to days 13–14, testing to days 15–17, and release and handover to day 18.

The project duration increases from 17 to 18 working days, even though the scope and total review effort remain unchanged.

Possible responses include obtaining a second qualified reviewer, securing additional availability or agreeing a revised completion date. Each option has implications for cost, feasibility and delivery commitments.

Resource levelling and resource smoothing

Two scheduling techniques help address resource demand:

Resource levelling adjusts activity dates to resolve resource limitations. It may change the critical path and extend project duration.

Resource smoothing adjusts activities within available float while preserving the critical path and planned completion date.

Moving Polish review to days 11–12 in our example is resource levelling. Smoothing alone cannot resolve the shortage: the two review activities require 16 person-hours across days 9–11, while the shared reviewer has only 12 available hours during that period.

These techniques show why resource planning and scheduling must be developed together. A mathematically correct activity network can still be infeasible when resource limitations are considered.

Choosing how to obtain resources

The team must also decide which resources can be provided internally and which require external support.

Internal staff may offer knowledge of the university's terminology and procedures. External specialists may provide additional capacity or competencies that are unavailable internally. The decision should consider relevant expertise, confirmed availability, coordination requirements and the total cost of obtaining an acceptable result.

For an external language service, the brief should specify the content volume, deliverables, review responsibilities, file formats, deadlines and acceptance criteria. This allows the team to compare offers on a consistent basis.

A lower hourly or per-word rate does not necessarily produce a lower total project cost if additional briefing, correction or coordination is required.

Estimating project costs

A *cost estimate* is an assessment of the resources needed to complete the agreed work, expressed in monetary terms. *Budgeting* brings these estimates together and establishes how funding will be allocated and controlled.

For labour estimated by the hour:

Labour cost = Estimated person-hours x Hourly cost rate

For example:

24 hours x €20=€480

This gives the estimated cost of Ukrainian translation under the stated assumptions.

Other cost calculations may use different units:

- Translation services may be priced per word or per agreed package.
- Software may be charged per user and subscription period.
- Specialist testing may have a fixed fee.
- Technical services may be charged by hours or usage.

The unit and coverage of each rate must be clear. For a per-word translation quote, the team should establish which word count is used and whether editing, terminology work and corrections are included.

Methods introduced in the previous section also apply to costs. Comparable projects support analogous estimates; quantity multiplied by a unit rate supports parametric estimates; estimating individual work packages supports bottom-up budgeting.

Including the full resource cost

A budget should cover all agreed work, including requirements clarification, project coordination, terminology preparation, review, testing and handover.

Staff time also has a resource cost when employees already receive salaries. An organisation may allocate part of that cost to the project even when no additional salary payment is required. A project's resource cost and its additional cash payments can therefore differ.

Shared administrative or infrastructure costs may also be allocated to the project. The team should specify whether these costs are included in the hourly rates or added separately, so that they are counted consistently.

Existing software may require no additional purchase, but the team must still confirm that the necessary access and licence capacity are available.

Running example: Building the cost estimate

The following estimate uses the *original 17-working-day plan* with separate language reviewers. All figures are hypothetical teaching assumptions expressed in euros.

The hourly rates represent the project's assumed labour cost, including the overhead included in this example. No additional overhead is added. Review and correction hours cover the planned review cycle; the testing allowance covers normal testing and correction work.

Work package or resource	Basis of estimate	Estimated cost
1.1 Project management and requirements	30 hours × €25	€750
1.2 Glossary and style guide	12 hours × €25	€300
1.3.1 Ukrainian content package	Translation: 24 hours × €20; review and correction: 8 hours × €25	€680
1.3.2 Polish content package	Translation: 30 hours × €20; review and correction: 8 hours × €25	€800
1.4.1 Integration	12 hours × €30	€360
1.4.2 Testing and corrections	18 hours × €25	€450
1.5 Release and handover	6 hours × €25	€150

Shared software and testing environment	Software: €120; testing environment: €80	€200
Total estimated cost before reserves		**€3,690**

The WBS references connect costs to the work they support. They also make later comparisons easier: if the Polish content package exceeds its estimate, the team can examine the hours or rates associated with that package.

The project management allowance covers coordination throughout the project. It is additional work beyond the delivery activities shown in the simplified network diagram.

Planning contingency reserves

Cost estimates contain uncertainty. A *contingency reserve* provides an allowance for identified risks affecting the agreed project scope. Its amount should reflect the relevant risks, their possible consequences and the organisation's approach to uncertainty.

For the portal project, consider two identified scenarios:

- Unexpected terminology ambiguities could require an additional specialist review: 8 hours × €25 = €200.
- Unexpected compatibility problems could require technical work beyond the planned correction allowance: 4 hours × €30 = €120.

For our running example, assume the sponsor decides to provide for the full estimated additional cost of both scenarios:

$$\textit{Contingency reserve} = €200 + €120 = €320$$

This is an explicit planning decision. It is not a universal percentage or a guarantee that every possible problem has been covered.

The ordinary review and correction work is already included in the base estimate. The reserve addresses additional work arising from the specified uncertainties, avoiding duplication of planned costs.

A financial reserve also does not automatically provide extra working days or ensure that a specialist will be available. Cost, resource and schedule responses must be considered together.

Distinguishing the cost baseline from the total project budget

The *cost baseline* is the approved budget distributed over time and used as a reference for monitoring cost performance. It includes contingency reserves but excludes management reserves.

A *management reserve* is an additional amount held for unforeseen work within the agreed project scope. Access to it follows the organisation's approval arrangements. When its use is approved, the relevant amount is incorporated into the cost baseline through the change process.

For our example, the proposed totals are:

Budget component	Amount
Estimated cost of planned work and resources	€3,690
Contingency reserve	€320
Proposed cost baseline total	**€4,010**
Illustrative management reserve	€200
Proposed total project budget	**€4,210**

The €200 management reserve is an additional teaching assumption. The appropriate amount and approval authority would depend on the organisation and project.

A request to add another language introduces new scope. It requires an assessment of the additional work and an authorised change to the relevant plans and funding. The existence of a reserve does not itself authorise that expansion.

Connecting the budget to time and funding

Before approval, the budget must be connected to the schedule. The team needs to identify when costs are expected to arise: terminology preparation occurs early, translation and review follow, and integration and testing occur later.

This is called *time-phasing the budget*. It provides a basis for comparing planned expenditure with actual costs at different points in the project.

The team should also identify when payments are due. Software access may require payment before work begins, while an external reviewer may invoice after acceptance. A sufficient total budget must therefore be supported by funding available at the required times.

Resource and budget decisions should be assessed against the agreed results. Additional reviewer capacity may protect the release date; reducing the review allowance may weaken the project's ability to meet its acceptance criteria. The project manager needs to explain these consequences when presenting alternatives.

Practical task

The planned Polish reviewer becomes unavailable. A replacement can perform the same eight hours of review within the original schedule but charges €35 per hour

instead of €25. Calculate the revised estimated cost before reserves, cost baseline total and total project budget, keeping both reserves unchanged. Explain what the project manager should confirm before accepting the replacement.

4. Project Risk and Quality Planning

A project plan depends on assumptions about the work, available resources and the conditions under which the team will operate. Risk planning examines what could affect those assumptions. Quality planning establishes how the team will produce and verify results that meet the agreed requirements.

For our admissions portal project, completing the work within 17 working days and the approved budget would be insufficient if applicants received inaccurate information. Planning must therefore address both uncertainty during delivery and the quality of the final portal.

Understanding project risks

A *project risk* is an uncertain event or condition that could affect one or more project objectives. A negative effect creates a *threat*, while a positive effect creates an *opportunity*.

For example, the Polish reviewer might become unavailable during the planned review period, threatening the completion date. Alternatively, the university might have a previously approved translation memory containing reusable material, creating an opportunity to reduce translation effort.

A *risk* concerns uncertainty. An *issue* is an existing situation requiring attention.

"The reviewer may become unavailable" describes a risk. "The reviewer has confirmed that they cannot work on the required dates" describes an issue. Similarly, the shared-reviewer overallocation examined in the previous section is a known planning problem once the resource limitation has been confirmed.

Risks require monitoring and planned responses. Existing issues require decisions and action.

Identifying and describing risks

Risk identification begins with reviewing the scope, requirements, schedule, resources and assumptions. The team can use previous project experience, stakeholder discussions, technical checks and specialist judgement.

For a language-related project, useful questions include:

- Are the source materials complete, approved and sufficiently clear?
- Could terminology require additional clarification?
- Are reviewers and client representatives available when needed?
- Could file formats or platform limitations affect integration?
- Could existing language resources reduce the required effort?

A useful risk statement connects a *cause*, an *uncertain event* and its *possible effect*.

For example:

Because several admissions terms have different institutional meanings, translators may select non-equivalent terms, resulting in misleading information or additional review work.

This is more actionable than a general label such as “translation risk”. It helps the team identify both preventive measures and the people who should be involved.

Assessing probability and impact

Identified risks need to be prioritised so that the team can direct attention and resources appropriately.

Probability describes how likely the event is to occur.

Impact describes its potential effect on objectives if it occurs. Impact may concern the schedule, costs, quality, scope or the usefulness of the result.

A small project can use qualitative ratings such as *low, medium and high*. The team should agree what these ratings mean and use available evidence consistently. A probability–impact matrix can then support prioritisation.

The following is an illustrative matrix for threats in our portal project.

Probability	Low impact	Medium impact	High impact
Low	Low priority	Low priority	High priority
Medium	Low priority	Medium priority	High priority
High	Medium priority	High priority	High priority

For this example, high impact includes publishing incorrect information about eligibility or deadlines, or jeopardising the agreed launch date. The matrix deliberately gives such consequences high priority even when their probability is low.

The ratings are planning judgements, not precise measurements. They should be reviewed when new information becomes available. A risk approaching its possible occurrence may also require more urgent attention than an equally rated risk later in the project.

Selecting risk responses

The response should address the specific risk and be feasible within the project’s authority and resources.

Common responses to threats include:

- *Avoid:* change the approach to remove the particular threat. For example, use a supported file format to avoid a risk arising specifically from an unsupported format.
- *Mitigate*: reduce the probability or impact. Approving terminology before translation can reduce inconsistent interpretation.

•*Transfer*: allocate specified responsibilities or financial consequences to another party through an appropriate agreement. The project team still needs to oversee delivery.

•*Accept:* acknowledge the risk and decide how it will be monitored and handled if it occurs. Acceptance may include a contingency plan.

•*Escalate:* refer the risk to an appropriate authority when it falls outside the project team's scope or decision-making powers.

For opportunities, the team may *exploit* the opportunity by acting to secure the benefit, *enhance* its likelihood or value, *share* it with a suitable partner, or *accept* it without taking special action.

In the portal project, the team could investigate whether approved translation memories contain relevant, current material. Reuse should be verified before anticipated savings are incorporated into the schedule or budget.

Recording risks and assigning ownership

A *risk register* records identified risks, their assessment, owners and planned responses. It should remain useful throughout the project.

The following extract shows three illustrative threats. The ratings describe their initial assessment, before considering the effectiveness of planned responses.

Risk and potential effect	Probability / impact	Owner and planned response
R1 Terminology ambiguity: inconsistent interpretation may produce misleading content or additional review work.	Medium / High	**Language lead:** clarify key terms with the admissions office and maintain an approved glossary.
R2 Compatibility problems: importing localised content may produce character or layout errors and require additional technical work.	Medium / Medium	**Technical lead:** test representative strings before completing the full import and retain recoverable source files.
R3 Reviewer unavailability: loss of Polish review capacity may delay activities on the critical path.	Low / High	**Project manager:** confirm the booking and identify a qualified replacement with suitable availability.

A fuller register can also include triggers, response deadlines, status, review dates and the risk remaining after the response.

The *risk owner* is responsible for monitoring the risk and coordinating the response. Other team members may carry out individual response actions.

A *trigger* is an observable condition that signals the need for action. For R3, a trigger might be failure to reconfirm the reviewer's availability by an agreed checkpoint. This gives the project manager time to investigate before the review activity is due to begin.

Connecting responses to resources and reserves

A response must specify what will be done, who will do it, when it will happen and how it will be resourced. A statement such as “use another reviewer if necessary” is incomplete until the team has considered competence, availability, access to materials and cost.

It is also useful to distinguish:

- Preventive or mitigating action: work undertaken to reduce exposure before the risk occurs.
- Contingency action: a prepared response activated when an agreed condition is met or the risk occurs.

In the previous section, the contingency reserve included:

- €200 for eight additional hours of specialist review arising from unexpected terminology difficulties.
- €120 for four additional hours of technical work arising from compatibility problems.

These allowances total €320. They cover the specified additional work beyond the planned review and correction activities.

For R2, the contingency response could involve restoring affected content from recoverable files, correcting the import problem and repeating the relevant tests. The technical lead would coordinate the work and report any expected effect on the schedule.

Preventive work also requires time and resources. The team must confirm whether it fits within existing activity estimates or requires an agreed revision. The financial reserve alone does not guarantee that the necessary specialist will be available.

After a response, some uncertainty may remain. This is *residual risk*. A response may also introduce a new risk: replacing a reviewer, for example, may create a need for additional terminology briefing. These effects should be considered when selecting the response.

Defining quality for the project

Quality concerns whether the deliverables meet their agreed requirements and are suitable for their intended use. For the admissions portal, this includes accurate information, consistent terminology, understandable language and correct operation in the target environment.

A translation can be grammatically correct while giving applicants the wrong deadline. Equally, an accurate translation can become unusable if essential text is missing or incorrectly displayed after integration.

For language quality, frameworks such as *Multidimensional Quality Metrics (MQM)* distinguish dimensions including accuracy, terminology, linguistic conventions, style and locale conventions. These categories can help a team describe errors consistently and select checks relevant to its project. A full scoring system is unnecessary for our introductory example.

Quality planning, assurance and control

Three related functions should be distinguished.

Quality planning establishes the relevant criteria, methods, responsibilities and evidence. It determines what will be checked, how and at which points.

Quality assurance (QA) provides confidence that the working processes can meet the quality requirements. In our project, this could include checking that the agreed review workflow is followed, reviewers use the correct source versions, and terminology decisions are communicated consistently.

Quality control (QC) examines the actual deliverables through review, measurement or testing. Examples include comparing a translation with the source, checking glossary compliance and testing the integrated portal. QA and QC are connected activities that support quality throughout delivery.

Automated checks can help identify missing strings, inconsistent terminology or damaged formatting. Human review remains necessary to assess meaning, context and suitability for applicants.

Turning acceptance criteria into a quality plan

The acceptance criteria developed in the first section now need corresponding checks, evidence and responsibilities.

Quality requirement	Planned check and evidence	Responsible role
Complete coverage: each language contains the agreed 20 webpages and 40 interface messages.	Compare delivered content with the approved inventory; record missing or duplicated items.	Language reviewer
Accurate admissions information: no unresolved errors alter eligibility, deadlines or required documents.	Compare target content with the approved source; record review findings and admissions clarification.	Language reviewer and admissions representative
Consistent terminology and style: content follows the approved glossary and style guide.	Run terminology checks and conduct editorial review; document approved exceptions.	Language lead and reviewer
Correct display: no missing characters, clipped text or overlapping labels remain in the agreed tests.	Test the integrated portal in the agreed browsers and screen sizes; retain test results and defect records.	Technical tester
Complete handover: required content files, language resources and documentation are available in the agreed formats.	Use a handover checklist and record receipt and acceptance.	Project manager and authorised client representative

The quality plan should also specify when checks occur.

In our schedule, terminology preparation is addressed in activity B, linguistic review in E and F, and integrated portal testing in H. Approval and handover are

completed in I. Checks within these activities should support timely correction before work passes to the next stage.

Classifying defects and deciding priorities

A *defect* is an observed failure to meet a requirement. Once detected, it requires correction or an authorised decision consistent with the agreed acceptance arrangements.

A simple project-specific severity classification could distinguish:

- *Critical defects:* errors that could seriously mislead applicants, such as an incorrect eligibility condition or application deadline.
- *Major defects:* substantial problems affecting content or use, such as a missing required page or broken navigation to essential information.
- *Minor defects:* limited presentation or language problems that do not change meaning or prevent use.

The team should agree these definitions before review begins. The categories support prioritisation; they do not override acceptance criteria. A defect classified as minor may still need correction because a specific requirement prohibits it.

A defect record should identify the affected item, the requirement not met, severity, responsible person and correction status. After correction, the relevant check should be repeated.

A single combined quality score can conceal an important error. For this project, an otherwise strong translation would still fail acceptance if it contained an unresolved error that changed an application deadline.

Considering the cost of quality

Quality-related costs include prevention, appraisal and the consequences of failures.

Terminology clarification and a reliable review process help prevent errors. Reviewing translations and testing the portal are appraisal activities. Correcting errors before handover creates internal failure costs; correcting them after publication creates external failure costs and may also affect users.

This distinction helps explain why quality work belongs in the original plan and budget. Removing a review activity reduces its immediate cost but may increase later correction work. The appropriate allocation depends on the consequences of defects and the agreed quality requirements.

Preparing for acceptance

Quality control produces evidence about whether requirements have been met. *Acceptance* is the authorised client decision to approve a deliverable.

The project team should therefore establish who can accept the language packages and final portal, which records support that decision, and how unresolved findings will

be handled. Acceptance can occur for individual deliverables as well as at final handover.

Risk and quality planning remain connected during delivery. Review findings may reveal risks that were underestimated, while a resource change may require adjustments to the planned quality process. The monitoring arrangements discussed in the next section will help the team recognise and respond to these developments.

Practical task

For a project creating an English–Ukrainian terminology database, identify two threats and one opportunity. For each, specify its possible effect, owner and response. Then select one deliverable, formulate two measurable acceptance criteria and explain how, when and by whom they will be checked.

5. Monitoring Project Progress and Managing Changes

A project plan establishes what the team intends to deliver, how the work will be organised, and which resources will be required. During implementation, the project manager needs evidence of what is actually happening and whether the agreed objectives remain achievable.

In translation, localisation and language technology projects, progress can be difficult to judge. A translation may be complete while terminology review is still pending. A localised webpage may pass a linguistic check but display incorrectly after integration. Monitoring therefore needs to cover both the work performed and the condition of the deliverables.

Monitoring and control

Project monitoring involves collecting and analysing information about performance.

Project control uses that information to identify necessary action, make decisions and follow up on their results. Together, they connect the approved plan with the work being carried out.

Three reference points are particularly useful:

- *Baseline:* the approved reference against which performance is assessed.
- *Actual performance:* what has happened up to a specified reporting date.
- *Forecast:* the current prediction of future performance, based on available evidence.

For our university admissions portal project, assume that the plans developed in the previous sections have been approved:

Planning element	Approved reference
Scope	Localise 20 webpages and 40 interface messages into each of two languages: Ukrainian and Polish
Schedule	Complete the project by the end of working day 17
Cost baseline	€4,010, including €320 contingency reserve
Management reserve	€200 outside the cost baseline
Acceptance requirements	Agreed linguistic, functional and display criteria

All figures in this example are illustrative.

The baseline provides a stable reference. The forecast changes as the team learns more about the work. If completion is now expected on day 18, the project manager records that forecast and explains the difference from the approved day-17 finish.

Choosing meaningful progress information

Useful monitoring information answers a practical question: *Are we likely to deliver the agreed result within the approved constraints?*

For a language-related project, a compact monitoring system can cover the following areas:

Area	Information to collect	Example of supporting evidence
Deliverables	Work completed and its current stage	Translation files, review records, accepted content
Schedule	Actual dates, remaining duration and dependencies	Updated activity schedule
Costs	Costs incurred and estimated remaining costs	Timesheets, invoices and updated estimates
Quality	Defects, correction status and acceptance results	Linguistic review and functional test records
Risks and issues	New concerns, unresolved problems and assigned actions	Updated risk and issue records
Changes	Requests received and decisions made	Change log and approval records

Progress measures should reflect the nature of the work. A page count is useful for tracking coverage, but pages can differ substantially in length and complexity. Similarly, the percentage of the budget spent does not show the percentage of the project completed.

For localisation, it is helpful to distinguish stages such as:

translated → reviewed → integrated → tested → accepted

Each stage needs clear evidence. For example, a reviewed translation requires a completed review and resolution of the relevant comments.

The team should agree who supplies each update and when. A small project might use brief daily coordination and a weekly status report, with significant problems reported as soon as they become apparent.

Worked example: updating the schedule

Consider the project at the *end of working day 9*, using the original arrangement with separate reviewers for Ukrainian and Polish.

The team reports:

- Source content, the glossary and the style guide are complete.
- Ukrainian translation finished on day 8.
- Ukrainian review started on day 9 and is expected to finish on day 10.
- Polish translation has covered 16 of 20 webpages and 32 of 40 interface messages.
- The remaining Polish translation is estimated to require one additional working day, involving six person-hours.

Polish translation was scheduled to finish on day 9. Its revised finish is day 10.

Because Polish translation lies on the critical path, the delay affects the activities that follow it. Assuming the original dependencies, resource availability and remaining activity durations still apply, the forecast becomes:

Activity	Baseline timing	Current forecast
Finish Polish translation	Day 9	Day 10
Review Polish translation	Days 10–11	Days 11–12
Integrate both language versions	Days 12–13	Days 13–14
Test and correct the portal	Days 14–16	Days 15–17
Approve, release and hand over	Day 17	Day 18

The project therefore has a f*orecast completion delay of one working day*.

The translated content counts show coverage within the Polish translation activity. They do not establish that the Polish version or the whole project is 80% complete. Review, integration, testing and acceptance still require work.

A useful schedule update records actual progress, reassesses remaining work and recalculates the consequences for completion. Changes to the approved schedule baseline follow the applicable change control procedure.

Monitoring costs and forecasting completion

Cost monitoring becomes more useful when spending to date is considered alongside the cost of the remaining work.

A basic forecast adds actual cost to an updated estimate to complete:

Estimate at Completion = Actual Cost + Estimate to Complete

This approach requires a fresh assessment of what remains to be done, using current information.

For the day-9 update, assume the following:

Cost measure	Amount
Actual cost incurred to date	€2,030
Updated estimate of remaining identified work	€1,780
Forecast cost of currently identified work	**€3,810**
Original estimate for planned work, before reserves	€3,690
Approved cost baseline, including contingency	€4,010

The forecast for identified work has increased by €120. In this example, that increase comes from six additional Polish translation hours at €20 per hour.

The forecast leaves:

€4,010-€3,810=€200 within the approved cost baseline.

This calculation excludes any further costs arising from remaining uncertainty. The project manager must therefore assess whether the remaining €200 is adequate for the risks still present and whether the proposed use of contingency is within their authority. The separate €200 management reserve remains subject to its approval arrangements.

When preparing the remaining-cost estimate, include outstanding work and relevant commitments once. For example, an unpaid reviewer invoice already recorded as an incurred cost should not also appear as future review work.

Taking action and reporting the project's status

A variance needs an explanation and a response proportionate to its effect.

For the Polish translation delay, the project manager should first establish the cause. Unresolved terminology, underestimated effort and insufficient availability require different responses.

Possible actions might include securing timely subject-matter advice, increasing a qualified reviewer's availability, or assessing whether independently reviewed content batches can be integrated earlier. Each option needs a check of its cost, dependencies and quality implications.

A recovery action should have:

- a responsible person;
- a completion date;
- an expected effect;
- a way to check whether it worked.

For example:

The project manager will confirm the Polish reviewer's availability by day 10 and assess whether the review can finish earlier while retaining the full planned review effort.

After that decision, the schedule must be recalculated. Faster completion of an activity improves the project finish only if the relevant dependencies and resource constraints allow it.

A status report brings this information together for decision-making. A concise report should cover progress, deviations, forecasts, important issues, next actions and decisions required.

For our example:

Status date: End of working day 9.

Progress: Ukrainian translation is complete and review is underway. Polish translation requires one additional day.

Forecast: Completion on day 18 against the approved day-17 finish. Currently identified work is forecast to cost €3,810 against the €4,010 cost baseline; remaining risk exposure requires reassessment.

Next action: The project manager will assess a schedule recovery option and its resource implications by day 10.

Decision required: If recovery is impractical, the sponsor must consider a revised completion commitment through the agreed approval process.

The report gives the reader enough information to understand the situation and act.

Managing requests for change

During implementation, stakeholders may request new content, different functionality or revised priorities. Some changes improve the usefulness of the final result. Their effects still need to be assessed before the team commits to delivering them.

Change control is the process for capturing, evaluating and deciding on requests that affect an approved baseline. A request may be approved, rejected or deferred. When approved, the relevant plans are updated and implementation is monitored.

The project should define which decisions the team or project manager can make and which require approval from the client, sponsor or another authorised person.

Consider these examples:

Situation	Appropriate treatment
A translation contains the wrong application deadline	Record and correct the defect against the existing requirement
A reviewer changes an inaccurate term to the approved glossary term	Complete the planned review and correction work
The client requests a German version of the portal	Assess a scope change
The client requests release three working days earlier	Assess the effects on the schedule, resources, costs and risks

A correction can still require escalation if its effects exceed the project manager's authority or threaten an approved commitment.

Scope creep occurs when additional work enters the project without adequate assessment and agreement. Recording requests makes their consequences visible before they become informal commitments.

A proportionate change procedure usually includes:

1. Record the request: describe the proposed change, its purpose, requester and date.
2. Assess its effects: consider deliverables, effort, resources, schedule, costs, quality and risks.
3. Obtain an authorised decision: approve, reject or defer, with a recorded reason.

4.Update the relevant plans: reflect approved changes in requirements, responsibilities and affected baselines.

5.Implement and verify: complete the work, check the result and close the request.

A deferred request also needs a review date or a condition for reconsideration.

Worked example: adding another language

Suppose the client asks:

*"*Could we also provide the admissions portal in German?"

This adds another language version to the agreed scope. The impact assessment should examine the complete delivery process.

Area	Questions for the assessment
Scope	Does German cover the same 20 webpages and 40 interface messages?
Language resources	Who will translate and review the content? What terminology and style guidance must be added?
Schedule	Which activities can run in parallel, and which depend on shared specialists or systems?
Costs	What additional translation, review, coordination, integration and testing effort is required?
Quality and acceptance	Who will verify the German content and authorise acceptance?
Risks	What uncertainty arises from terminology, resource availability or display requirements?

Adding a third language does not automatically increase total project cost or duration by 50 %. Some preparation can be reused, some work can run in parallel, and some shared activities may become bottlenecks.

The project manager can present options: include German in the current release with revised commitments, deliver it in a later phase, or defer the request. Until a decision is approved, it remains a proposed change and should be shown separately from the committed scope.

After approval, the team updates the affected records and identifies the new versions clearly. The change record should connect the decision with its date, approver, rationale and resulting commitments.

The distinction between *updating a forecast* and *approving a new baseline* remains essential. A forecast reports what is currently expected. An approved baseline revision records an authorised change to the project's commitments.

Practical task

Use the day-9 information from the portal localisation project.

1.Write a short status report covering completed work, the schedule deviation, the cost forecast and the next action.

2.The client requests five additional FAQ pages in each target language. Identify the information needed to assess this request.

3.Recommend whether to approve, reject or defer the request. State your assumptions and explain the expected effects on scope, time, cost and quality.

I. Review Questions

1. What is the difference between product scope and project scope in a localisation project?
2. How do requirements and acceptance criteria help define and verify project deliverables?
3. How do the WBS, work packages and the 100% rule help ensure that all agreed project work is covered?
4. How does effort differ from duration, and how does resource availability affect duration estimates?
5. What do an activity network diagram and a Gantt chart show, and how do they support project scheduling?
6. How are the critical path and total float determined, and what do they tell us about the consequences of delays?
7. How do resource levelling and resource smoothing differ in their effects on the project schedule?
8. Which costs should be included in the budget of a translation or localisation project?
9. How do contingency reserves and management reserves relate to the cost baseline and the total project budget?
10. How does a risk differ from an issue, and why can project risks include both threats and opportunities?
11. What information should a risk register contain, and how does it support risk prioritisation and response planning?
12. How do quality assurance, quality control and client acceptance differ?
13. What is the difference between a baseline, actual performance and a forecast when monitoring project progress?
14. How can actual costs and an updated estimate to complete be used to forecast the final project cost?
15. How should the team assess and manage a request to add another language to the admissions portal, and how does change control prevent scope creep?

II. Key Terms for Independent Study. Define each term in your own words and provide an example from an applied linguistics project.

1. Project scope
2. Work Breakdown Structure (WBS)
3. Critical path
4. Total float
5. Resource levelling
6. Contingency reserve
7. Risk register

8. Quality assurance (QA)
9. Baseline
10. Change control