STATUS ACCOUNTING REPORT

Status Accounting Process

This document ensures a structured way to track and manage changes in the WebCongress project. It defines the process, data, and reporting methods necessary to keep clear records of all modifications, issues, and progress during the project lifecycle.

DATA FOR STATUS ACCOUNTING

This section outlines the essential data that must be collected to evaluate, document, and monitor any change introduced in the WebCongress system. Each type of data serves a specific purpose in project management and configuration control.

Change Requests (CRs)

Data to collect:

- CR ID: Unique identifier for the change.
- Description: What is being changed or added.
- Priority: Importance level (High, Medium, Low).
- Estimated Time: Approximate duration to complete the change.
- Budget Impact: Effect on the current project budget.
- Required Skills: Technical knowledge needed to perform the change.
-

CR Class (Type of Change)

- Functional enhancement
- UI improvement
- Bug fix
- Security update

CI Reference (Configuration Item Reference)

- Requirements documentation
- Design specifications
- Code commits or repositories
- Previous baseline versions

Baseline Updates

- New version number
- Summary of feature changes
- Associated release notes

Persons Assigned

- Name of personnel assigned
- Role (Frontend, Backend, QA, etc.)
- Skill level (Jr, Mid, Sr)

Maintenance Status

- Integrated
- Partially integrated
- Not integrated

Time Tracking

- Total estimated time
- Time elapsed so far
- Time spent on coding
- Time allocated to testing
- Time for optimization or deployment

Budget

- Estimated budget for the change
- Actual budget spent so far

Infrastructure Used

- Servers or cloud services
- Tools and frameworks used

Issues Encountered

- Description of the issue
- Associated risk
- Severity level

RULES FOR STATUS ACCOUNTING

Rule 1: If the time is over and the implementation is not finished, a report of the main issues found, the context, and the current outcomes should be provided.

Rule 2: If the budget is lower than estimated, the unfinished features should be reported, also the percentage of features completed.

Rule 3: If there are features non-implemented after the estimated time and budget, an analysis about what required skills are missing in the developers' team must be presented.

Rule 4: Weekly project meetings must be held and documented to ensure visibility of progress and blockers.

Rule 5 (Final Report): Upon completion, a closing report with charts comparing planned vs. actual time and budget, issue log summaries, and lessons learned must be submitted.

PROCESS FOR STATUS ACCOUNTING

Faster implementation: Implementation is completed before the estimated time and/or under budget. Possible causes may include high team efficiency or lower task complexity. Actions include documenting success strategies for future CRs.

Planned implementation: Implementation is completed as scheduled and within budget. All features are delivered as planned. Actions include updating final reports and validating compliance with requirements.

Slower implementation: Implementation exceeds time and/or budget. Some features may remain pending. Causes can include unexpected complexity, skill gaps, or technical difficulties. Actions include applying status rules, identifying planning issues, and adjusting timelines or scope.

STATUS ACCOUNTING REPORT FORMAT

1. **Date of CR approval**
2. **CR Description and ID**
3. **CI References** (requirements, design, code, previous baseline): versions
4. **Involved people** (names and roles)
5. **Estimated budget**
6. **Estimated time**
7. **Estimated people** (roles required)
8. **Performed budget**
9. **Performed time**
10. **Performance of human resources**
11. **Issues:**
    a. Description of the issue
    b. Author
    c. Involved personnel
    d. Risks

e. Possible solutions
f. Outcome (solved, unsolved, addressing) and rationale

12. **Outcome of the implementation of the CR**

Scale for tracking

1. Time Tracking Scale

Ahead of Schedule: >10% earlier than planned
On Schedule: Within ±10% of estimated time
Delayed: More than 10% beyond estimated time

2. Budget Performance Scale

Under Budget: Spent <90% of allocated budget
On Budget: Spent between 90%–110% of allocated budget
Over Budget: Spent >110% of allocated budget

3. Human Resource Performance Scale

High Efficiency: Tasks completed on time or earlier, high code quality
Moderate Efficiency: Minor delays, acceptable quality
Low Efficiency: Delays >20%, multiple revisions required

4. Issue Outcome Scale

Solved: Fully resolved, no further action needed
Addressing: Being worked on, partial mitigation in place
Unsolved: Still blocking progress, unresolved risk

5. CR Impact Priority

- **High**: Affects critical system functionality or delivery deadlines
- **Medium**: Affects non-core functionality or UI
- **Low**: Cosmetic or minor enhancement