# Daily Workflow - Quick Reference Guide

**Version**: 1.1 - Simplified
**Last Updated**: 2025-11-11

---

## 🌅 Start of Day (10 minutes)

### 1. Check What Happened Yesterday (3 min)

bash

```bash
cd /path/to/ConstructionPlatform
git log --oneline -5
```

**Read these files:**

- `docs/sprints/sprint-01/PROGRESS.md` (last section) → What did I do?
- `docs/sprints/sprint-01/PLAN.md` (today's section) → What's today's plan?

**Write down 1-3 goals for today.**

### 2. Log Start Time (2 min)

Open: `docs/time-tracking/2025-11-week*.md`

Add:

markdown

```markdown
### Session N
- **Start**: HH:MM
- **Planned Work**: [What you're doing today]
```

---

## 💻 During Work

### Work in Small Chunks

- Code for 20-30 minutes
- Commit your changes
- Repeat

### Commit Often

bash

```
git add .
git commit -m "feat: what I just did"
git push
```

### If You Hit a Blocker

1. Write it down in `docs/sprints/sprint-01/PROGRESS.md`
2. Try to fix it for 30 minutes
3. If still stuck, document it and move on

### If You Make an Important Decision

Write it in `docs/sprints/sprint-01/DECISIONS.md`:

markdown

```
### [Date] What I Decided

**Decision**: What I chose
**Why**: Reason for choosing it
```

---

## 🌙 End of Day (10 minutes)

### 1. Log Your Time (2 min)

Update `docs/time-tracking/2025-11-week*.md`:

markdown

```
- **End**: HH:MM
- **Duration**: X.X hours
- **Activity**: What I worked on
- **Category**: Planned | Debugging | Unplanned
```

### 2. Update What You Did (5 min)

Update `docs/sprints/sprint-01/PROGRESS.md`:

markdown

```markdown
## Day N: [Date]

### Work Completed
- ✅ Did this thing
- ✅ Fixed this problem
- 🚧 Started this (not finished)

### Blockers
- ⚠️ Problem I ran into: [description]
```

### 3. Final Commit (3 min)

bash

```bash
git add .
git commit -m "docs: End of day N - [summary]"
git push
```

---

## 📅 Friday Review (30 minutes)

### Calculate Your Speed

Look at `docs/time-tracking/2025-11-week*.md`

- How many hours did you plan?
- How many hours did you actually work?
- Divide: actual ÷ planned = your velocity

### Review Problems

Look at `docs/sprints/sprint-01/PROGRESS.md`

- What blockers did you hit?
- Did you solve them?
- What did you learn?

### Plan Next Week

Look at `docs/sprints/sprint-01/PLAN.md`

- What's coming up?
- Adjust time estimates based on this week's velocity

---

## 📂 The 3 Files You Use Every Day

| File | When | What |
|---|---|---|
| **PLAN.md** | Morning | Today's tasks |
| **PROGRESS.md** | Evening | What you did |
| **Time Tracking** | Start & End | Hours worked |

**Location**: `/path/to/ConstructionPlatform/docs/sprints/sprint-01/`

---

## 🆘 Emergency Procedures

### Session Interrupted?

bash

```
git add .
git commit -m "WIP: Interrupted - [what you were doing]"
git push
```

### Forgot to Track Time?

bash

```
# Check your commits to estimate time
git log --since="today" --pretty=format:"%h %ad | %s" --date=format:'%H:%M'
```

#### Lost Your Place?

1. Read last commit message
2. Read `PROGRESS.md` today's section
3. Check `git diff HEAD~3..HEAD`

---

## ⏱️ Time Investment

**Daily**: 20 minutes total

- Morning: 10 minutes
- Evening: 10 minutes

**Weekly**: 30 minutes

- Friday review

**Benefit**: Never lose context, always know what to do next

---

## ✅ Simple Checklist

### Morning

- □ Check yesterday's progress (3 min)
- □ Read today's plan (3 min)
- □ Log start time (2 min)
- □ Write 1-3 goals (2 min)

### During Day

- □ Code for 20-30 minutes
- □ Commit changes
- □ Repeat

### Evening

- □ Log end time (2 min)
- □ Update what you did (5 min)
- □ Final commit and push (3 min)

### Friday

- □ Calculate velocity (5 min)
- □ Review blockers (10 min)
- □ Plan next week (5 min)

---

## 💡 Golden Rules

1. **Commit every 20-30 minutes** - Small chunks are easier to track
2. **Never skip end-of-day** - 10 minutes saves hours later
3. **Write blockers immediately** - Don't forget what went wrong
4. **Keep it simple** - If it's too complex, you won't do it

---

## 🎯 Today's Quick Start

**Right now, do this:**

1. Open `docs/sprints/sprint-01/PROGRESS.md` → Check yesterday
2. Open `docs/sprints/sprint-01/PLAN.md` → Check today's tasks
3. Open `docs/time-tracking/2025-11-week*.md` → Log start time
4. Write down 1-3 specific goals for today
5. Start coding

**That's it. Get to work!** 🚀

---

**Remember**: The goal is progress, not perfect documentation. Do the minimum to stay organized, then focus on building.

---

## 📋 Daily Session Template (Copy This)

DATE: _____________ SESSION: _____

START TIME: _______ END TIME: _______

TODAY'S GOALS:
1. _________________________________
2. _________________________________
3. _________________________________

WORK COMPLETED:
□ _________________________________
□ _________________________________
□ _________________________________

BLOCKERS ENCOUNTERED:
_____________________________________
_____________________________________

DECISIONS MADE:
_____________________________________
_____________________________________

NEXT SESSION PRIORITIES:
1. _________________________________
2. _________________________________

DURATION: _______ hours
CATEGORY: □ Planned □ Debugging □ Unplanned

---

# 🗂️ Document Locations Reference

### Daily Use Documents

📋
✓

```
ConstructionPlatform/
├── docs/
│   ├── sprints/
│   │   └── sprint-01/
│   │       ├── PLAN.md          ← Today's tasks
│   │       ├── PROGRESS.md      ← Work log
│   │       └── DECISIONS.md     ← Important choices
│   └── time-tracking/
│       └── 2025-11-week*.md     ← Hours tracking
```

### Reference Documents

```
ConstructionPlatform/
├── docs/
│   ├── technical-debt/
│   │   └── REGISTER.md          ← Problems to fix
│   ├── validation prompts/
│   │   └── IMPLEMENTATION_VALIDATION_CHECKLIST.md
│   └── CHANGELOG.md             ← Feature completions
```

---

## 📞 Quick Commands Cheat Sheet

### Morning Startup

bash

```bash
cd /path/to/ConstructionPlatform
git status
git log --oneline -5
cat docs/sprints/sprint-01/PROGRESS.md | tail -50
```

### During Work

bash

```
# Commit frequently
git add .
git commit -m "feat: description"
git push

# Check if tests pass
npm test

# Check TypeScript compilation
npm run build
```

### End of Day

bash

```
# Final commit with summary
git add .
git commit -m "docs: End of day N - summary"
git push
```

### Emergency Save

bash

```
git add .
git commit -m "WIP: Interrupted"
git push
```

---

**Print this guide and keep it at your desk!**

For the full detailed workflow, see: `Daily_Workflow_Full_Version.md`